# 32-bit RISC Microcontroller
# Reference manual

# 32-bit Timer Event Counter
# (T32A-C)

# Revision 1.2

**2024-11**

**Toshiba Electronic Devices & Storage Corporation**

# Contents

# List of figures

# List of tables

## Preface

**Related Documents**

| Document name |
| --- |
| Datasheet |
| Exception |
| Clock Control and Operation Mode |
| Product Information |

## Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal:     0xABC
  Decimal:         123 or 0d123        - Only when it needs to be explicitly shown that they are decimal numbers.
  Binary:          0b111               - It is possible to omit the "0b" when the number of bits can be distinctly understood from a sentence.

- "_N" is added to the end of signal names to indicate low active signals.

- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.

- When two or more signal names are referred, they are described like as [m:n].
  Example: S[3:0] shows four signal names S3, S2, S1 and S0 together.

- The characters surrounded by *[ ]* defines the register.
  Example: *[ABCD]*

- "N" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: *[XYZ1]*, *[XYZ2]*, *[XYZ3]* → *[XYZn]*

- "x" substitutes suffix number or character of units and channels in the register list.

- In case of unit, "x" means A, B, and C, ...
  Example: *[ADACR0]*, *[ADBCR0]*, *[ADCCR0]* → *[ADxCR0]*

- In case of channel, "x" means 0, 1, and 2, ...
  Example: *[T32A0RUNA]*, *[T32A1RUNA]*, *[T32A2RUNA]* → *[T32AxRUNA]*

- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.

- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: *[ABCD]*<EFG> = 0x01 (hexadecimal), *[XYZn]*<VW> = 1 (binary)

- Word and byte represent the following bit length.
  Byte:              8 bits
  Half word:        16 bits
  Word:             32 bits
  Double word:      64 bits

- Properties of each bit in a register are expressed as follows:
  R:        Read only
  W:        Write only
  R/W:      Read and write are possible.

- Unless otherwise specified, register access supports only word access.

- The register defined as "Reserved" must not be rewritten. Moreover, do not use the read value.

- The value read from the bit having default value of "-" is unknown.

- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.

- Reserved bits of the write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.

- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

**Terms and Abbreviations**

Some of abbreviations used in this document are as follows:

DMA       Direct Memory Access
PPG       Programmable Pulse Generator
T32A       32-bit Timer Event Counter

# 1. Outline

T32A can work as a 16-bit timer (timer A, timer B) of 2ch or a 32-bit timer (timer C) of 1ch by 1UNIT circuit unit.

| Function classification | | | Function | | Functional description or range |
|---|---|---|---|---|---|
| 16-bit timer<br><br>Timer A | Time control | | Interval timer | | This function generates interrupt to CPU at every set interval time. |
| | Measurement control | | Event counter | | Up counting or down counting or up/down counting can be selected as count operation. And when the count value matches the timer register, you can generate timer interrupt. |
| | | | Capture | Frequency measurement | Captures count value on rising edge and falling edge of input pulse. You can calculate frequency from difference of captured data. |
| | | | | Pulse width measurement | Captures count value on rising edge and falling edge of input pulse. You can calculate Pulse Width from difference of captured data. |
| | | | | Time difference measurement | Captures count value on rising edge or falling edge of input pulse. You can calculate Time difference from difference of captured data. |
| | Rectangular wave output | | PPG | | Can output rectangular wave of arbitrary frequency and arbitrary duty. |
| | Synchronous operation | | Counter start | | Timer counter starts the count in synch with the start of master timer counter. |
| | | | Counter stop | | Timer counter stops the count in synch with the stop of master timer counter. |
| | | | Counter reload | | Timer counter is reloaded in synch with the of master timer counter. |
| 16-bit timer<br><br>Timer B | Time control | | Interval timer | | This function generates interrupt to CPU at every set interval time. |
| | Measurement control | | Event counter | | Up counting or down counting or up/down counting can be selected as count operation. And when the count value matches the timer register, you can generate timer interrupt. |
| | | | Capture | Frequency measurement | Captures count value on rising edge and falling edge of input pulse. You can calculate frequency from difference of captured data. |
| | | | | Pulse width measurement | Captures count value on rising edge and falling edge of input pulse. You can calculate Pulse Width from difference of captured data. |
| | | | | Time difference measurement | Captures count value on rising edge or falling edge of input pulse. You can calculate Time difference from difference of captured data. |
| | Rectangular wave output | | PPG | | Can output rectangular wave of arbitrary frequency and arbitrary duty. |
| | Synchronous operation | | Counter start | | Timer counter starts the count in synch with the start of master timer counter. |
| | | | Counter stop | | Timer counter stops the count in synch with the stop of master timer counter. |
| | | | Counter reload | | Timer counter is reloaded in synch with the of master timer counter. |

| Function classification | | Function | | Functional description or range |
|---|---|---|---|---|
| 32-bit timer<br><br>Timer C | Time control | Interval timer | | This function generates interrupt to CPU at every set interval time. |
| | Measurement control | Event counter | | Up counting or down counting or up/down counting can be selected as count operation. And when the count value matches the timer register, you can generate timer interrupt. |
| | | Pulse count | 1-phase pulse count | Counts the variation in the inputs of T32AxINC0 or T32AxINC1. The counter increment or decrement is selected according to the setting. |
| | | | 2-phase pulse count | Increment or decrement of counter according to the variation in the combination of inputs T32AxINC0 and T32AxINC1. |
| | | Capture | Frequency measurement | Captures count value on rising edge and falling edge of input pulse. You can calculate frequency from difference of captured data. |
| | | | Pulse width measurement | Captures count value on rising edge and falling edge of input pulse. You can calculate Pulse Width from difference of captured data. |
| | | | Time difference measurement | Captures count value on rising edge or falling edge of input pulse. You can calculate Time difference from difference of captured data. |
| | Rectangular wave output | PPG | | Can output rectangular wave of arbitrary frequency and arbitrary duty. |
| | Synchronous operation | Counter start | | Timer counter starts the count in synch with the start of master timer counter. |
| | | Counter stop | | Timer counter stops the count in synch with the stop of master timer counter. |
| | | Counter reload | | Timer counter is reloaded in synch with the of master timer counter. |

# 2. Configuration

T32A consists of two 16-bit timers, which operate as Timer A and Timer B respectively. Timer A and Timer B are connected, then they operate as 32-bit timer C.

## 2.1. 16-bit Timer

In 16-bit timer, the T32A is configured with the independent 16-bit Timer A and Timer B.
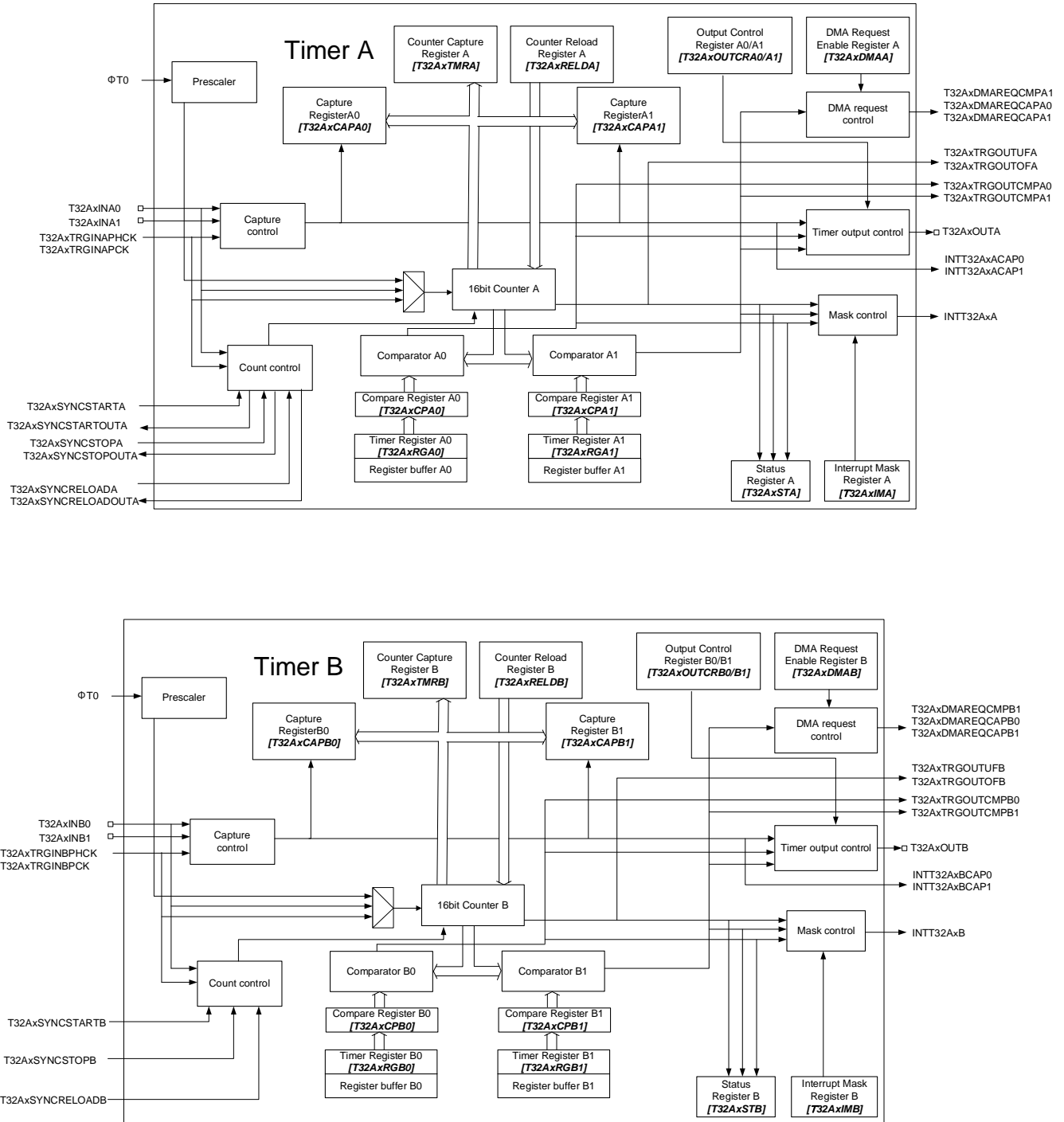


**Figure 2.1　Block Diagram of Timer A and Timer B in 16-bit Timer**

**Table 2.1　16-bit Timer Connection Specification**

| No. | Symbol | Signal name | I/O | Related reference manual |
|---|---|---|---|---|
| 1 | ΦT0 | Prescaler clock | Input | Clock Control and Operation Mode |
| 2 | T32AxINA0 | Timer A external trigger input 0 | Input | Datasheet |
| 3 | T32AxINA1 | Timer A external trigger input 1 | Input | Datasheet |
| 4 | T32AxTRGINAPHCK | Timer A trigger from other timer | Input | Product Information |
| 5 | T32AxTRGINAPCK | Timer A internal trigger input | Input | Product Information |
| 6 | T32AxSYNCSTARTA | Timer A synchronous start at trigger input | Input | Product Information |
| 7 | T32AxSYNCSTARTOUTA | Timer A trigger output for synchronous start | Output | Product Information |
| 8 | T32AxSYNCSTOPA | Timer A synchronous stop at trigger input | Input | Product Information |
| 9 | T32AxSYNCSTOPOUTA | Timer A trigger output for synchronous stop | Output | Product Information |
| 10 | T32AxSYNCRELOADA | Timer A synchronous reload at trigger input | Input | Product Information |
| 11 | T32AxSYNCRELOADOUTA | Timer A trigger output for synchronous reload | Output | Product Information |
| 12 | T32AxDMAREQCMPA1 | DMA request at match A1 register | Output | Product Information |
| 13 | T32AxDMAREQCAPA0 | DMA request at capture A0 register | Output | Product Information |
| 14 | T32AxDMAREQCAPA1 | DMA request at capture A1 register | Output | Product Information |
| 15 | T32AxTRGOUTUFA | Timer A underflow trigger | Output | Product Information |
| 16 | T32AxTRGOUTOFA | Timer A overflow trigger | Output | Product Information |
| 17 | T32AxTRGOUTCMPA0 | Timer register A0 match trigger | Output | Product Information |
| 18 | T32AxTRGOUTCMPA1 | Timer register A1 match trigger | Output | Product Information |
| 19 | T32AxOUTA | Timer A output | Output | Datasheet/ Product Information |
| 20 | INTT32AxACAP0 | Timer A capture register 0 interrupt | Output | Exception |
| 21 | INTT32AxACAP1 | Timer A capture register 1 interrupt | Output | Exception |
| 22 | INTT32AxA | Timer A match, overflow and underflow interrupt | Output | Exception |
| 23 | T32AxINB0 | Timer B external trigger input 0 | Input | Datasheet |
| 24 | T32AxINB1 | Timer B external trigger input 1 | Input | Datasheet |
| 25 | T32AxTRGINBPHCK | Timer B trigger from other timer | Input | Product Information |
| 26 | T32AxTRGINBPCK | Timer B internal trigger input | Input | Product Information |
| 27 | T32AxSYNCSTARTB | Timer B synchronous start at trigger input | Input | Product Information |
| 28 | T32AxSYNCSTOPB | Timer B synchronous stop at trigger input | Input | Product Information |
| 29 | T32AxSYNCRELOADB | Timer B synchronous reload at trigger input | Input | Product Information |
| 30 | T32AxDMAREQCMPB1 | DMA request at match B1 register | Output | Product Information |
| 31 | T32AxDMAREQCAPB0 | DMA request at capture B0 register | Output | Product Information |
| 32 | T32AxDMAREQCAPB1 | DMA request at capture B1 register | Output | Product Information |
| 33 | T32AxTRGOUTUFB | Timer B underflow trigger | Output | Product Information |
| 34 | T32AxTRGOUTOFB | Timer B overflow trigger | Output | Product Information |
| 35 | T32AxTRGOUTCMPB0 | Timer register B0 match trigger | Output | Product Information |
| 36 | T32AxTRGOUTCMPB1 | Timer register B1 match trigger | Output | Product Information |
| 37 | T32AxOUTB | Timer B output | Output | Datasheet/ Product Information |
| 38 | INTT32AxBCAP0 | Timer B capture register 0 interrupt | Output | Exception |
| 39 | INTT32AxBCAP1 | Timer B capture register 1 interrupt | Output | Exception |
| 40 | INTT32AxB | Timer B match, overflow and underflow interrupt | Output | Exception |

## 2.2. 32-bit Timer

In 32-bit timer, the T32A is configured with the 32-bit Timer C.
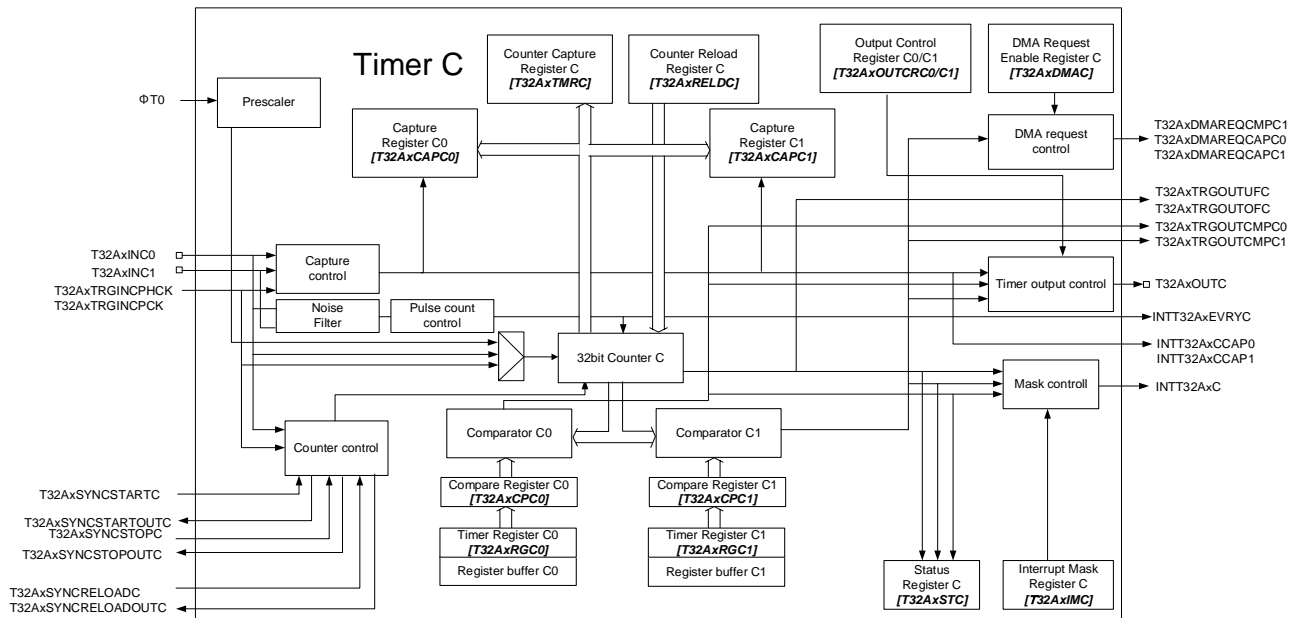


**Figure 2.2   Block Diagram of Timer C in 32-bit Timer**

**Table 2.2   32-bit Timer Connection Specification**

| No. | Symbol | Signal name | I/O | Related reference manual |
|---|---|---|---|---|
| 1 | ΦT0 | Prescaler clock | Input | Clock Control and Operation Mode |
| 2 | T32AxINC0 | Timer C external trigger input 0 | Input | Datasheet |
| 3 | T32AxINC1 | Timer C external trigger input 1 | Input | Datasheet |
| 4 | T32AxTRGINCPHCK | Timer C trigger from other timer | Input | Product Information |
| 5 | T32AxTRGINCPCK | Timer C internal trigger input | Input | Product Information |
| 6 | T32AxSYNCSTARTC | Timer C synchronous start at trigger input | Input | Product Information |
| 7 | T32AxSYNCSTARTOUTC | Timer C trigger output for synchronous start | Output | Product Information |
| 8 | T32AxSYNCSTOPC | Timer C synchronous stop at trigger input | Input | Product Information |
| 9 | T32AxSYNCSTOPOUTC | Timer C trigger output for synchronous stop | Output | Product Information |
| 10 | T32AxSYNCRELOADC | Timer C synchronous reload at trigger input | Input | Product Information |
| 11 | T32AxSYNCRELOADOUTC | Timer C trigger output for synchronous reload | Output | Product Information |
| 12 | T32AxDMAREQCMPC1 | DMA request at match C1 register | Output | Product Information |
| 13 | T32AxDMAREQCAPC0 | DMA request at capture C0 register | Output | Product Information |
| 14 | T32AxDMAREQCAPC1 | DMA request at capture C1 register | Output | Product Information |
| 15 | T32AxTRGOUTUFC | Timer C underflow trigger | Output | Product Information |
| 16 | T32AxTRGOUTOFC | Timer C overflow trigger | Output | Product Information |
| 17 | T32AxTRGOUTCMPC0 | Timer register C0 match trigger | Output | Product Information |
| 18 | T32AxTRGOUTCMPC1 | Timer register C1 match trigger | Output | Product Information |
| 19 | T32AxOUTC | Timer C output | Output | Datasheet/ Product Information |
| 20 | INTT32AxCCAP0 | Timer C capture register 0 interrupt | Output | Exception |
| 21 | INTT32AxCCAP1 | Timer C capture register 1 interrupt | Output | Exception |
| 22 | INTT32AxC | Timer C match, overflow and underflow interrupt | Output | Exception |
| 23 | INTT32AxEVRYC | Every count interrupt | Output | Exception |

# 3. Function and Operation

T32A is composed two 16-bit timer that can be used Timer A and Timer B. Also it can use Timer C that is connected Timer A and Timer B as 32-bit timer. When the Timer C is used, Timer A and Timer B cannot be used.

The 16-bit timer and the 32-bit timer mode setting are specified as follows:

| Operation mode |
| --- |
| *[T32AxMOD]*<MODE32> |

## 3.1. Clock Supply

When T32A is used, the corresponding clock enable bits should be set to "1" (Clock supply) in fsys supply stop register A (*[CGFSYSENA]* and *[CGFSYSMENA]*), fsys supply stop register B (*[CGFSYSENB]* and *[CGFSYSMENB]*), fsys supply stop register C(*[CGFSYSMENC]*), and fc supply stop register (*[CGFCEN]*).

The corresponding registers and the bit locations depend on a product. Some products do not have all registers. For the details, refer to "Clock Control and Operation Mode" of reference manual.

## 3.2. Prescaler

The prescaler divides the prescaler clock ($\Phi$T0) to generate the source clock for the counter.

In 16-bit timer, the division ratio can be selected for Timer A and Timer B respectively. In 32-bit timer, the division ratio can be selected with Timer C. The division ratio is specified as follows:

| Prescaler division ratio selection bits | |
| --- | --- |
| Timer A | *[T32AxCRA]*<PRSCLA> |
| Timer B | *[T32AxCRB]*<PRSCLB> |
| Timer C | *[T32AxCRC]*<PRSCLC> |

For the setting of $\Phi$T0, refer to reference manual "Clock Control and Operation Mode".

## 3.3. Counters (16-bit Counter A/16-bit Counter B/32-bit Counter C)

In 16-bit timer, the Timer A/Timer B operates as 16-bit counters respectively. In 32-bit timer, the Timer C operates as a 32-bit counter.

### 3.3.1. Count Clock

As the count clock of Timer A, Timer B, or Timer C, it can be selected from a prescaler output, an output of other timer, an external trigger (T32AxINA0/T32AxINB0/T32AxINC0), or an internal trigger. The count clock can be selected from the bits in the table below.

| Count clock selection bits | |
|---|---|
| Timer A | *[T32AxCRA]*<CLKA> |
| Timer B | *[T32AxCRB]*<CLKB> |
| Timer C (Note) | *[T32AxCRC]*<CLKC> |

Note: When pulse count operation in 32-bit timer is selected (*[T32AxCRC]* <UPDNC> = 11), the setting above is ignored.

### 3.3.2. Starting Operation

When "1" is set to the timer operation control bit of each timer, the value in the Counter Reload Register is reloaded (initial reloading). When the start conditions are established, the timer operation starts.

| Counter reload register | |
|---|---|
| Timer A | *[T32AxRELDA]* |
| Timer B | *[T32AxRELDB]* |
| Timer C | *[T32AxRELDC]* |

| Timer operation control | |
|---|---|
| Timer A | *[T32AxRUNA]*<RUNA> |
| Timer B | *[T32AxRUNB]*<RUNB> |
| Timer C | *[T32AxRUNC]*<RUNC> |

When "1" is set to Timer operation control bits, make sure that the timer is stopping. About timer stopping, please refer to "3.3.3 Stopping Operation".

The counter operation is started by a register setting using program, internal triggers, external triggers, output from other timer, or channel synchronous with other channels.

When the pulse counting in 32-bit timer is used, an external trigger cannot be used as the factor to start the counter.

The factor of starting the counter is specified with the counter start condition setting bits.

| Counter start condition setting bits | |
|---|---|
| Timer A | *[T32AxCRA]* <STARTA> |
| Timer B | *[T32AxCRB]* <STARTB> |
| Timer C | *[T32AxCRC]* <STARTC> |

To start the T32A by program, set "000" to the counter start condition setting bits.

Below describe factors of count operation start.

(1)  Timer operation start by program

When "1" is set to "starts the operation by program" in table below, the counter starts operation.

| Starts operation by program | |
|---|---|
| Timer A | *[T32AxRUNA]*<SFTSTAA> |
| Timer B | *[T32AxRUNB]*<SFTSTAB> |
| Timer C | *[T32AxRUNC]*<SFTSTAC> |

(2)  Internal trigger

The counter starts operation by an internal trigger from other peripheral function. The internal trigger is selected outside of the T32A. Do not select the same internal trigger to start and stop the counter.

(3)  External trigger

The counter starts operation on the rising/falling edge of an external signal (T32AxINA0, T32AxINB0, or T32AxINC0). Do not select the same edge for starting the counter and stopping the counter.

(4)  Timer output from other timer

The counter starts operation on the rising/falling edge of the output signal from other timer. Do not select the same edge for starting the counter and stopping the counter.

(5)  Synchronous start

The counter can start synchronously with start of other timer. Synchronous start setting should be specified to the slave timer. For more information, refer to "3.4 synchronous operation".

### 3.3.3. Stopping Operation

To check whether the counter is operating or stopping of each timer, use the operation flag of timer.

| Operation flag of timer | |
|---|---|
| Timer A | *[T32AxRUNA]*<RUNFLGA> |
| Timer B | *[T32AxRUNB]*<RUNFLGB> |
| Timer C | *[T32AxRUNC]*<RUNFLGC> |

The counter operation is stopped by register setting using program, internal triggers, external triggers, outputs from others timers, channel synchronous with other channels, or compare match between the counter and Timer Register 1.

| Timer register 1 | |
|---|---|
| Timer A | *[T32AxRGA1]* |
| Timer B | *[T32AxRGB1]* |
| Timer C | *[T32AxRGC1]* |

When the pulse counting in timer C is used (*[T32AxCRC]*<UPDNC> = 11), an external trigger cannot be used as the factor to stop the counter.

The factor of stopping the counter is specified with the counter stop condition setting bits.

| Counter stop condition setting bits | |
|---|---|
| Timer A | *[T32AxCRA]* <STOPA> |
| Timer B | *[T32AxCRB]* <STOPB> |
| Timer C | *[T32AxCRC]* <STOPC> |

To stop the T32A by program, set "000" to the counter stop condition setting bits.

Below describe factors of count operation start/stop

(1) Timer operation stop by program

When "1" is set to "Stop the operation by program" in table below, the counter stops operation.

| Stop operation by program | |
|---|---|
| Timer A | *[T32AxRUNA]*<SFTSTPA> |
| Timer B | *[T32AxRUNB]*<SFTSTPB> |
| Timer C | *[T32AxRUNC]*<SFTSTPC> |

(2) Internal trigger

The counter stops operation by an internal trigger from other peripheral function. The internal trigger is selected outside of the T32A. Do not select the same internal trigger to start and stop the counter.

(3) External trigger

The counter stops operation on the rising/falling edge of an external signal (T32AxINA0, T32AxINB0, or T32AxINC0). Do not select the same edge for starting the counter and stopping the counter.

(4) Signal output from other timer

The counter stops operation on the rising/falling edge of the signal output from the other timer. Do not select the same edge for starting the counter and stopping the counter.

(5) Synchronous start

The counter can stop synchronously with start of other timer. Synchronous stop setting should be specified to the slave timer. For more information, see "3.4 synchronous operation".

(6) Compare match between the counter and Timer Register 1

The counter stops when a compare match between the counter and Timer Register 1 is detected.

| Timer register 1 | |
|---|---|
| Timer A | *[T32AxRGA1]* |
| Timer B | *[T32AxRGB1]* |
| Timer C | *[T32AxRGC1]* |

### 3.3.4. Count Operation

Up counting or down counting or up/down counting can be selected as the counter operation.

| Counter operation select | |
|---|---|
| Timer A | *[T32AxCRA]* <UPDNA> |
| Timer B | *[T32AxCRB]* <UPDNB> |
| Timer C | *[T32AxCRC]* <UPDNC> |

When up/down counting is specified, after counter starts, if the reload condition is established, the counter is reloaded. The counter repeats up or down counting until the stop condition is established. For reload condition, refer to "3.3.7 Reloading Counter"

When up/down counting is specified, the count starts up and counts until the counter matches with Timer Register 1. After the match, the counter operation changes to down counting until the counter value reaches "0". Then the counter operation changes to up counting. The counter repeats this operation until the stop condition is established.

| Timer register 1 | |
|---|---|
| Timer A | *[T32AxRGA1]* |
| Timer B | *[T32AxRGB1]* |
| Timer C | *[T32AxRGC1]* |

To check whether the counter is operating or stopping, use the operation flag of timer.

| Operation flag of timer | |
|---|---|
| Timer A | *[T32AxRUNA]*<RUNFLGA> |
| Timer B | *[T32AxRUNB]*<RUNFLGB> |
| Timer C | *[T32AxRUNC]*<RUNFLGC> |

In the counting operation at PPG output, ensure that the relationship in the following table holds for Timer Register 0 (*[T32AxRGA0]*/*[T32AxRGB0]* /*[T32AxRGC0]*), Timer Register 1 (*[T32AxRGA1]*/*[T32AxRGB1]*/ *[T32AxRGC1]*), and Counter Reload Register (*[T32AxRELDA]*/*[T32AxRELDB]*/*[T32AxRELDC]*)

**Table 3.1   Note of Timer Register Setting when Outputting PPG**

| Count operation | Setting condition |
|---|---|
| Up counting | Setting range of *[T32AxRGx1]*: *[T32AxRGx1]* ≥ *[T32AxRELDx]* +2<br>Setting range of *[T32AxRGx0]*: *[T32AxRELDx]* ≤ *[T32AxRGx0]* ≤ *[T32AxRGx1]*<br>Setting of output control register：set or clear |
| Down counting | Setting range of *[T32AxRGx1]*: *[T32AxRGx1]* ≤ *[T32AxRELDx]* – 2<br>Setting range of *[T32AxRGx0]*: *[T32AxRELDx]* ≥ *[T32AxRGx0]* ≥ *[T32AxRGx1]*<br>Setting of output control register：set or clear |
| Up/down counting | Setting range of *[T32AxRGx1]*: *[T32AxRGx1]* ≥ 2 AND *[T32xRGx1]* = *[T32xRELDx]*<br>Setting range of *[T32AxRGx0]*: 0 < *[T32AxRGx0]* < *[T32AxRGx1]*<br>Setting of output control register：reverse<br>Set to *[T32AxRGx1]* before set to *[T32AxRGx0]*. |

### 3.3.5. Event Count Operation

The event count operates in Timer A, Timer B and Timer C respectively.

Specifying external trigger, internal trigger and signal output from other timer as count clock, it can be event counter.

Select up counting as the counter operation. Count up by external trigger (On the rising edge or on the falling edge), signal output from the other timer (On the rising edge or on the falling edge) or internal trigger. You can read following register as count of events.

| Counter capture register | |
|---|---|
| Timer A | *[T32AxTMRA]* |
| Timer B | *[T32AxTMRB]* |
| Timer C | *[T32AxTMRC]* |

### 3.3.6. Pulse Counting

Pulse counting operates in Timer C.

When "11" is set to the counter operation bit, the T32A enters pulse count mode.

| Counter operation | |
|---|---|
| Timer C | *[T32AxCRC]* <UPDNC> |

After above setting, select 2-phase pulse count mode or 1-phase pulse count mode with as follows:

| Pulse counter mode | |
|---|---|
| Timer C | *[T32AxPLSCR]* <PMODE> |

In 1-phase pulse count mode, the counter is incremented or decremented by the change of input of T32AxINC0 or T32AxINC1.

In 2-phase pulse count mode, the counter is incremented or decremented by the combination of the inputs of T32AxINC0 and T32AxINC1.

When the counter overflows, the counter becomes "0x00000000"; when the counter underflow, the counter becomes "0xFFFFFFFF". In both cases, the counter continues counting and timer interrupt occurs.

Timer interrupt occurs by compare match timer register C. For detail about timer interrupt, refer to "3.9.1 Timer Interrupt".

A noise filter can be inserted to T32AxINC0/T32AxINC1 with the noise elimination time for T32AxINC0 /T32AxINC1 bit as follows:

| Noise elimination time for T32AxINC0/T32AxINC1 | |
|---|---|
| Timer C | *[T32AxPLSCR]* <NF> |

(1) 1-phase pulse count mode

This mode counts up or counts down according to the level of input pulse to T32AxINC0 or T32AxINC1.

Counter conditions in the up count and down count can be selected with count condition select bit.

| Count condition specified | |
|---|---|
| Up counting | *[T32AxPLSCR]*<PUP> |
| Down counting | *[T32AxPLSCR]*<PDN> |

Do not set the same value to <PUP> and <PDN>.

(2) 2-phase pulse count mode

In this mode the counter counts up or down according to the change of the signal level input to T32AxINC0/T32AxINC1. By the combination of input signal levels, there are the following four states. These four states show the signals input to T32AxINC0 / T32AxINC1 in hexadecimal notation.

**Table 3.2   2-phase Pulse Count Mode**

| T32AxINC1 | T32AxINC0 | STATE |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

The figure below shows the state transition from/to up counting to/from down counting.

| Direction of 2-phase pulse counter | |
|---|---|
| Timer C | *[T32AxPLSCR]* <PDIR> |
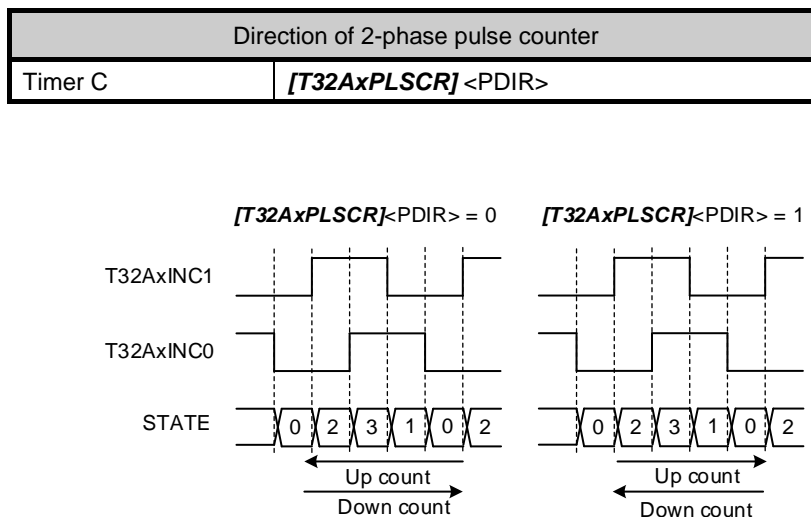


**Figure 3.1   State Transition**

When *[T32AxPLSCR]* <PDIR>=0 and the rising edge of T32AxINC0 changes before T32AxINC1, counter is up counting. When the rising edge of T32AxINC0 changes later than T32AxINC1 counter is down counting.

When *[T32AxPLSCR]* <PDIR> = 1 and the rising edge of T32AxINC1 changes before T32AxINC0, counter is down counting. When the rising edge of T32AxINC1 change later than T32AxINC0 counter is up counting.

The state transitions of $0 \rightarrow 3$, $3 \rightarrow 0$, $1 \rightarrow 2$, and $2 \rightarrow 1$, in which the counter is not changed, are state errors. These are the interrupt factors (INTT32AxC).

**Table 3.3   Transition**

| | Up counting | | Down counting | |
|---|---|---|---|---|
| | Before transition | After transition | Before transition | After transition |
| Positive phase <PDIR>=0 | 0 | 1 | 0 | 2 |
| | 1 | 3 | 2 | 3 |
| | 3 | 2 | 3 | 1 |
| | 2 | 0 | 1 | 0 |
| Negative phase <PDIR>=1 | 0 | 2 | 0 | 1 |
| | 2 | 3 | 1 | 3 |
| | 3 | 1 | 3 | 2 |
| | 1 | 0 | 2 | 0 |

(a)  Positive phase (Counter operation when <PDIR> = 0)
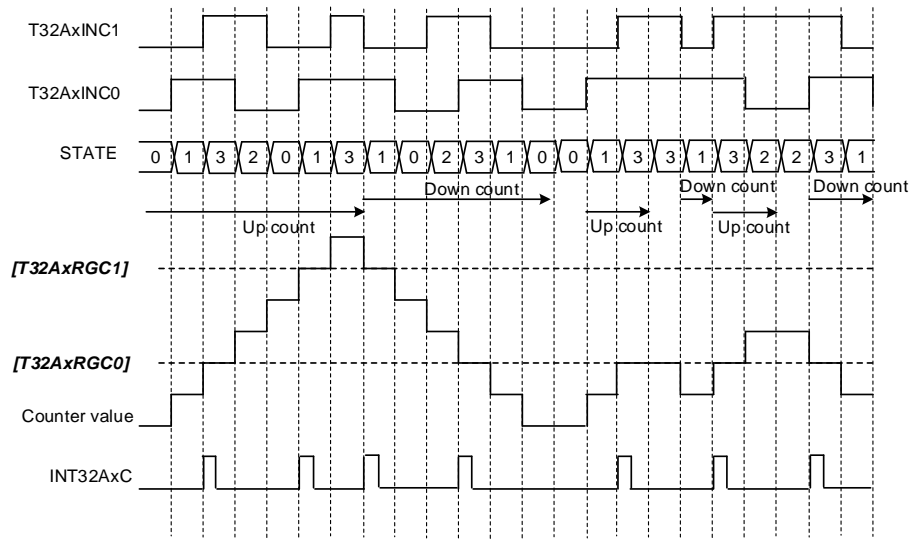


**Figure 3.2   Positive Phase (Counter Operation when <PDIR> = 0)**

(b) Negative phase (Counter operation when <PDIR> = 1)
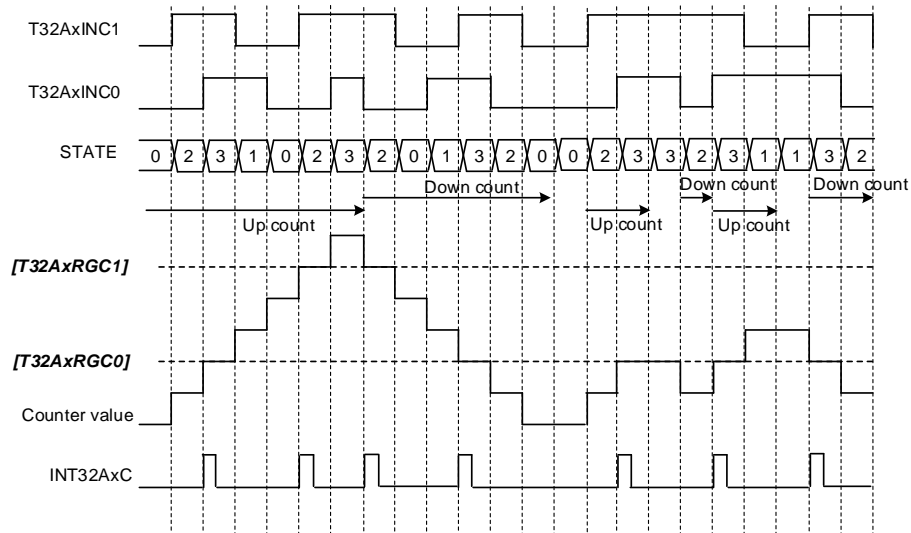


**Figure 3.3   Negative Phase (Counter Operation when <PDIR> = 1)**

## 3.3.7. Reloading Counter

You can re-setup the counter value at an arbitrary timing. The counter is reloaded by an internal trigger, external trigger, signal output from other timer, channel synchronous, compare match between the counter and Timer Register 1.

The reload condition can be set by the counter reload condition bit in the table below.

| Counter reload condition | |
|---|---|
| Timer A | *[T32AxCRA]* <RELDA**>** |
| Timer B | *[T32AxCRB]* <RELDB**>** |
| Timer C | *[T32AxCRC]* <RELDC**>** |

The reload value can be set by the Counter Reload Register in the table below.

| Counter reload register | |
|---|---|
| Timer A | *[T32AxRELDA]* |
| Timer B | *[T32AxRELDB]* |
| Timer C | *[T32AxRELDC]* |

When the reload value is set to "0", the counter is cleared. This can be used as counter clearing function.

The timer and Timer Register 1 are shown in the table below.

| Timer register 1 | |
|---|---|
| Timer A | *[T32AxRGA1]* |
| Timer B | *[T32AxRGB1]* |
| Timer C (Note) | *[T32AxRGC1]* |

Note: When the pulse count operation in Timer C is used, an external trigger cannot be used as the factor to reload the counter.

Counter reload condition is as following.

(1) compare match between the counter and Timer Register 1

The counter is reloaded when a compare match between the counter and Timer Register 1.

(2) External trigger

The counter is reloaded or cleared on the rising/falling edge of an external signal (T32AxINA0/T32AxINB0/T32AxINC0).

(3) Other timer outputs

The counter is reloaded or cleared on the rising/falling edge of a signal output from other timer.

(4) Internal trigger

The counter is reloaded by an internal trigger from other peripheral functions. An internal trigger is specified outside of the T32A.

(5) Channel synchronization

The counter can be synchronized with other timer's reloading. Synchronous start setting should be specified to the slave timer.

(6) No reloading

Set the counter as free-running without reloading.

## 3.4. Synchronous Operation

Multiple timers can synchronously be started, reloaded, and stopped. Timer A or Timer C as a master timer and multiple slaves can synchronously operate. For the channel connection of Synchronous Operation, refer to "Product Information" of reference manual.

Connection example in 16-bit mode          Connection example in 32-bit mode
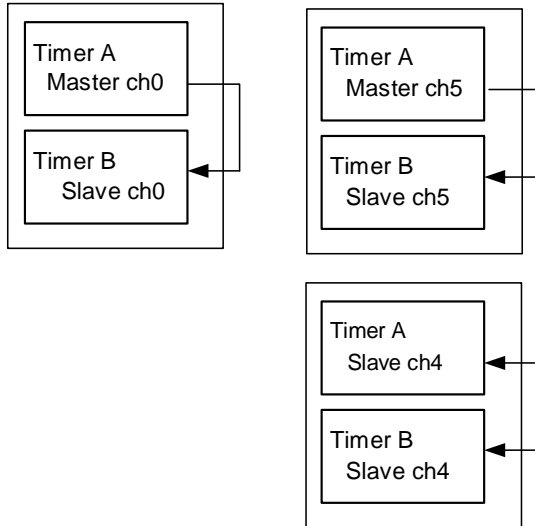
| Timer A Master ch0 |
| Timer B Slave ch0 |

| Timer A Master ch5 |
| Timer B Slave ch5 |

| Timer A Slave ch4 |
| Timer B Slave ch4 |

| Timer C Master ch0 |
| Timer C Slave ch1 |
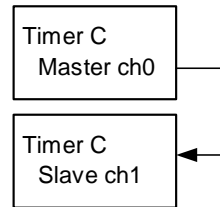
**Figure 3.4   Synchronous Operation**

## 3.5. Comparator

The comparator compares the value of the counter and the Compare Register. When a match is detected, a compare match signal is output. This compare match signal is a factor of interrupt requests or the factor to reverse the timer output. The compare match signal is also output as an internal trigger to other channels of the T32A or other peripheral functions.

| Compare register 1 | |
| --- | --- |
| Timer A | *[T32AxCPA1]* |
| Timer B | *[T32AxCPB1]* |
| Timer C | *[T32AxCPC1]* |

| Compare register 0 | |
| --- | --- |
| Timer A | *[T32AxCPA0]* |
| Timer B | *[T32AxCPB0]* |
| Timer C | *[T32AxCPC0]* |

## 3.6. Timer Register (*[T32AxRGA0/B0/C0], [T32AxRGA1/B1/C1]*)

The Timer Registers are used to set the compare value for the counter. Two registers are provided for Timer A, Timer B, and Timer C each.

The Timer Registers are configured in double-buffering and are paired with the register buffers respectively.

Double buffering is initially disabled.

When double buffering is disabled, the Timer Register operates as a single buffer. The value to compare is written to Timer Register 0/1directly for the counter.

When double buffering is enabled, if the counter matches with Timer Register 1, the value to compare is transferred from register buffer to Timer Register 1. And Timer Register 0 is updated at same time.

Using double buffering, you can update the value to compare regardless of update timing of the Timer Register.

| Timer register 1 | |
| --- | --- |
| Timer A | *[T32AxRGA1]* |
| Timer B | *[T32AxRGB1]* |
| Timer C | *[T32AxRGC1]* |

| Timer register 0 | |
| --- | --- |
| Timer A | *[T32AxRGA0]* |
| Timer B | *[T32AxRGB0]* |
| Timer C | *[T32AxRGC0]* |

Double buffering is enabled/disabled with double-buffering control bits in table below.

| Double-buffering control bit | |
| --- | --- |
| Timer A | *[T32AxCRA]*<WBFA> |
| Timer B | *[T32AxCRB]*<WBFB> |
| Timer C | *[T32AxCRC]*<WBFC> |

Note While the counter is stopping, even if double buffering is enabled, data is written to the Timer Register directly. This operation is single buffering.

## 3.7. Capture Control

The counter value can be captured at any timing. There are 2 ways as below.

**(1) Using external triggers, signal output from the other timer, or internal triggers**

A counter value is captured to Capture Register 0 and Capture Register 1 in the table below at the change of the level of an external trigger or signal output from other timer; or occurrence of an internal trigger.

| Capture register 0 | |
| --- | --- |
| Timer A | *[T32AxCAPA0]* |
| Timer B | *[T32AxCAPB0]* |
| Timer C | *[T32AxCAPC0]* |

| Capture register 1 | |
| --- | --- |
| Timer A | *[T32AxCAPA1]* |
| Timer B | *[T32AxCAPB1]* |
| Timer C | *[T32AxCAPC1]* |

The capture timing can be controlled by selecting capture timing bit shown in the tables below.

A capture timing can be selected from the following 7 timings: on the rising/falling edge of T32AxINA0/T32AxINB0/T32AxINC0, on the rising/falling edge of T32AxINA1/T32AxINB1/T32xAINC1, on the rising/falling edge of an signal output from other timer, or the occurrence of internal trigger.

| Capture timing sets bits of capture register 0 | |
| --- | --- |
| Timer A | *[T32AxCAPCRA]*<CAPMA0> |
| Timer B | *[T32AxCAPCRB]*<CAPMB0> |
| Timer C | *[T32AxCAPCRC]*<CAPMC0> |

| Capture timing sets bits of capture register 1 | |
| --- | --- |
| Timer A | *[T32AxCAPCRA]*<CAPMA1> |
| Timer B | *[T32AxCAPCRB]*<CAPMB1> |
| Timer C | *[T32AxCAPCRC]*<CAPMC1> |

**(2) Reading the counter**

When the Counter Capture Register in table below is read during the counter operation, current counter value can be captured and read. When the counter is stopping, the last captured value is maintained.

| Counter capture register | |
| --- | --- |
| Timer A | *[T32AxTMRA]* |
| Timer B | *[T32AxTMRB]* |
| Timer C | *[T32AxTMRC]* |

## 3.8. Output Programmable Rectangular Wave (PPG) (T32AxOUTA/T32AxOUTB/T32AxOUTC)

It can output rectangular wave of any frequency and any duty ratio. It can be set low active or high active as output pulse. It can output programmable rectangular wave at timer output terminal A/B/C when the counter and timer register match or when the counter is captured in capture register.

When a PPG output is used, specify the corresponding port setting in advance.

A "Low" signal is output at the initial state.

If multiple factors for setting, clearing, and reversing occur at the same time, a PPG output is not changed.

However, if all of the following conditions are established, the change caused by timer register 0 is priority.

- – When a match by timer register 0 and other conditions are established at the same time.
- – When control setting by a match by timer register 0 is set or cleared.
- – When timer register 0 and reload register have the same value.

### (1) Initial Output control

A PPG output can be set, cleared, or reversed with output control bit of Output Control Register 0 in table below. Writing data to this bit is always enabled. Even if the timer is stopping, the data is written to this bit.

| Control of T32AxOUTA/T32AxOUTB/T32AxOUTC | |
|---|---|
| Timer A | *[T32AxOUTCRA0]* <OCRA> |
| Timer B | *[T32AxOUTCRB0]* <OCRB> |
| Timer C | *[T32AxOUTCRC0]* <OCRC> |

### (2) PPG output when compare match

A PPG output can be set, cleared, or reversed when the counter matches with Timer Register.

It is selectable with compare control bit on Output Control Register 1 in table below.

| Control of T32AxOUTA/T32AxOUTB/T32AxOUTC by comparator | | | |
|---|---|---|---|
| Timer | Output control | Output control factor | Control register/symbol |
| Timer A | T32AxOUTA | Counter match with *[T32AxRGA0]* | *[T32AxOUTCRA1]*<OCRCMPA0> |
| | | Counter match with *[T32AxRGA1]* | *[T32AxOUTCRA1]*<OCRCMPA1> |
| Timer B | T32AxOUTB | Counter match with *[T32AxRGB0]* | *[T32AxOUTCRB1]*<OCRCMPB0> |
| | | Counter match with *[T32AxRGB1]* | *[T32AxOUTCRB1]*<OCRCMPB1> |
| Timer C | T32AxOUTC | Counter match with *[T32AxRGC0]* | *[T32AxOUTCRC1]*<OCRCMPC0> |
| | | Counter match with *[T32AxRGC1]* | *[T32AxOUTCRC1]*<OCRCMPC1> |
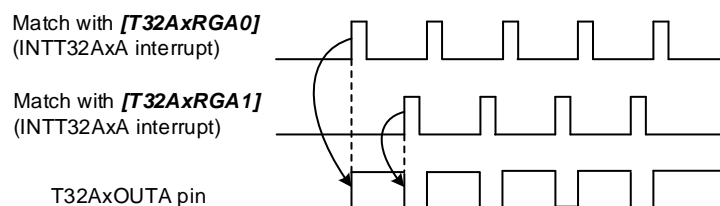


**Figure 3.5   Example of Output of Programmable Rectangular Wave Using 16-bit Timer A**

**(3) Capturing**

A PPG output can be set, cleared, or reversed with when counter is captured in *[T32AxCAPA0]*/*[T32AxCAPA1]*, *[T32AxCAPB0]*/*[T32AxCAPB1]* and *[T32AxCAPC0]*/*[T32AxCAPC1]*.

It is selectable with capture control bit on Output Control Register 1 in table below.

| Control of T32AxOUTA/T32AxOUTB/T32AxOUTC by capture register | | | |
|---|---|---|---|
| Timer | Output control | Output control factor | Control register/symbol |
| Timer A | T32AxOUTA | Capture to *[T32AxCAPA0]* | *[T32AxOUTCRA1]*<OCRCAPA0> |
| | | Capture to *[T32AxCAPA1]* | *[T32AxOUTCRA1]*<OCRCAPA1> |
| Timer B | T32AxOUTB | Capture to *[T32AxCAPB0]* | *[T32AxOUTCRB1]*<OCRCAPB0> |
| | | Capture to *[T32AxCAPB1]* | *[T32AxOUTCRB1]*<OCRCAPB1> |
| Timer C | T32AxOUTC | Capture to *[T32AxCAPC0]* | *[T32AxOUTCRC1]*<OCRCAPC0> |
| | | Capture to *[T32AxCAPC1]* | *[T32AxOUTCRC1]*<OCRCAPC1> |

## 3.9. Interrupts

The following three types of interrupt requests are output.

- Timer interrupt
  INTT32AxA, INTT32AxB, INTT32AxC

- Capture interrupt
  INTT32AxACAP0/1, INTT32AxBCAP0/1, INTT32AxCCAP0/1

- Interrupt on every count
  INTT32AxEVRYC

### 3.9.1. Timer Interrupt

Timer interrupts are INTT32AxA, INTT32AxB, and INTT32AxC. Timer interrupt is generated by factors in table below.

**Table 3.4   List of Timer Interrupt Factors**

| Timer | Interrupt name | Interrupt factor |
|---|---|---|
| Timer A | INTT32AxA | Counter match with *[T32AxRGA0]*, *[T32AxRGA1]* |
| | | Counter overflow |
| | | Counter underflow |
| Timer B | INTT32AxB | Counter match with *[T32AxRGB0]*, *[T32AxRGB1]* |
| | | Counter overflow |
| | | Counter underflow |
| Timer C | INTT32AxC | Counter match with *[T32AxRGC0]*, *[T32AxRGC1]* |
| | | Counter overflow |
| | | Counter underflow |

When Timer C is used in 2-phase pulse count, if the counter does not count up or count down, a state transition error (INTT32AxC) is output.

Each factor can be masked with Interrupt Mask Register in table below.

| Interrupt mask register | |
|---|---|
| Timer A | *[T32AxIMA]* |
| Timer B | *[T32AxIMB]* |
| Timer C | *[T32AxIMC]* |

Even if the mask register is enabled, the occurrence of the factor is set in Status Register in table below. To clear the state, write "1" to the corresponding bit.

| Status register | |
|---|---|
| Timer A | *[T32AxSTA]* |
| Timer B | *[T32AxSTB]* |
| Timer C | *[T32AxSTC]* |

### 3.9.2. Capture Interrupt

Capture interrupts are output when the counter value is captured to the capture register.

Capture interrupt and Capture Register are shown in the table below.

**Table 3.5    Relationship Table between Capture Interrupt and Capture Register**

| Relationship between of capture interrupt and capture register | | |
|---|---|---|
| Timer | Interrupt name | Capture register |
| Timer A | INTT32AxACAP0 | *[T32AxCAPA0]* |
| | INTT32AxACAP1 | *[T32AxCAPA1]* |
| Timer B | INTT32AxBCAP0 | *[T32AxCAPB0]* |
| | INTT32AxBCAP1 | *[T32AxCAPB1]* |
| Timer C | INTT32AxCCAP0 | *[T32AxCAPC0]* |
| | INTT32AxCCAP1 | *[T32AxCAPC1]* |

### 3.9.3. Interrupt on Every Count

An interrupt on every count (INTT32AxEVRYC) only occurs in pulse count mode (*[T32AxCRC]*<UPDNC> =11) with 32-bit mode. This interrupt is output on every up/down counting.

## 3.10. DMA Request

A DMA request is issued to the DMA controller when the counter matches the Timer Register 1 or the capture. When a DMA transfer is performed, enable the corresponding bit of the *[T32AxDMAA]* / *[T32AxDMAB]* / *[T32AxDMAC]* register.

| DMA request enable register | |
|---|---|
| Timer A | *[T32AxDMAA]* |
| Timer B | *[T32AxDMAB]* |
| Timer C | *[T32AxDMAC]* |

**Table 3.6    DMA Request List**

| Timer | DMA request | DMA request factor | Register |
|---|---|---|---|
| Timer A | T32AxDMAREQCMPA1 | Detect match between timer register A1(*[T32AxRGA1]*) and counter | *[T32AxDMAA]* <DMAENA2> |
| | T32AxDMAREQCAPA0 | Capture to capture register A0 ([*T32AxCAPA0]*) | *[T32AxDMAA]* <DMAENA0> |
| | T32AxDMAREQCAPA1 | Capture to capture register A1 (*[T32AxCAPA1]*) | *[T32AxDMAA]* <DMAENA1> |
| Timer B | T32AxDMAREQCMPB1 | Detect match between timer register B1(*[T32AxRGB1]*) and counter | *[T32AxDMAB]* <DMAENB2> |
| | T32AxDMAREQCAPB0 | Capture to capture register B0 (*[T32AxCAPB0]*) | *[T32AxDMAB]* <DMAENB0> |
| | T32AxDMAREQCAPB1 | Capture to capture register B1 (*[T32AxCAPB1]*) | *[T32AxDMAB]* <DMAENB1> |
| Timer C | T32AxDMAREQCMPC1 | Detect match between timer register C1 (*[T32AxRGC1]*) and counter | *[T32AxDMAC]* <DMAENC2> |
| | T32AxDMAREQCAPC0 | Capture to capture register C0 (*[T32AxCAPC0]*) | *[T32AxDMAC]* <DMAENC0> |
| | T32AxDMAREQCAPC1 | Capture to capture register C1 (*[T32AxCAPC1]*) | *[T32AxDMAC]* <DMAENC1> |

# 4. Registers

## 4.1. List of Registers

This section describes the control registers and their addresses.

There are common registers, Timer A registers, Timer B registers, and Timer C registers. Common registers are used in regardless of the mode. Timer A registers and Timer B registers are used in 16-bit timer. Timer C registers are used in 32-bit timer.

**(1) Base address**

| Function | | Channel/unit | Base address | | |
|---|---|---|---|---|---|
| | | | TYPE1 | TYPE2 | TYPE3 |
| 32-bit timer event counter | T32A | ch0 | 0x400BA000 | 0x400C1000 | 0x40061000 |
| | | ch1 | 0x400BA100 | 0x400C1400 | 0x40061400 |
| | | ch2 | 0x400BA200 | 0x400C1800 | 0x40061800 |
| | | ch3 | 0x400BA300 | 0x400C1C00 | 0x40061C00 |
| | | ch4 | 0x400BA400 | 0x400C2000 | 0x40062000 |
| | | ch5 | 0x400BA500 | 0x400C2400 | 0x40062400 |
| | | ch6 | 0x400BA600 | 0x400C2800 | 0x40062800 |
| | | ch7 | 0x400BA700 | 0x400C2C00 | 0x40062C00 |
| | | ch8 | 0x400BA800 | 0x400C3000 | 0x40063000 |
| | | ch9 | 0x400BA900 | 0x400C3400 | 0x40063400 |
| | | ch10 | 0x400BAA00 | 0x400C3800 | 0x40063800 |
| | | ch11 | 0x400BAB00 | 0x400C3C00 | 0x40063C00 |
| | | ch12 | 0x400BAC00 | 0x400C4000 | 0x40064000 |
| | | ch13 | - | 0x400C4400 | 0x40064400 |
| | | ch14 | - | 0x400C4800 | - |
| | | ch15 | - | 0x400C4C00 | - |

Note: The base address is different by products. Please refer to "Product Information" of the reference manual for the details.

**(2) Common register**

| Register name | | Address(Base+) |
|---|---|---|
| Mode Register | *[T32AxMOD]* | 0x0000 |

**(3) Timer A register (16-bit timer)**

| Register name | | Address(Base+) |
|---|---|---|
| RUN Register A | *[T32AxRUNA]* | 0x0040 |
| Counter Control Register A | *[T32AxCRA]* | 0x0044 |
| Capture Control Register A | *[T32AxCAPCRA]* | 0x0048 |
| Output Control Register A0 | *[T32AxOUTCRA0]* | 0x004C |
| Output Control Register A1 | *[T32AxOUTCRA1]* | 0x0050 |
| Status Register A | *[T32AxSTA]* | 0x0054 |
| Interrupt Mask Register A | *[T32AxIMA]* | 0x0058 |
| Counter Capture Register A | *[T32AxTMRA]* | 0x005C |
| Counter Reload Register A | *[T32AxRELDA]* | 0x0060 |
| Timer Register A0 | *[T32AxRGA0]* | 0x0064 |
| Timer Register A1 | *[T32AxRGA1]* | 0x0068 |
| Capture Register A0 | *[T32AxCAPA0]* | 0x006C |
| Capture Register A1 | *[T32AxCAPA1]* | 0x0070 |
| DMA Request Enable Register A | *[T32AxDMAA]* | 0x0074 |
| Compare Register A0 (Note) | *[T32AxCPA0]* | 0x0078 |
| Compare Register A1 (Note) | *[T32AxCPA1]* | 0x007C |

Note: When the base address is TYPE1, this register cannot be accessed.
For base address, refer to "Product Information" of reference manual.

**⑷ Timer B register (16-bit timer)**

| Register name | | Address(Base+) |
|---|---|---|
| RUN Register B | *[T32AxRUNB]* | 0x0080 |
| Counter Control Register B | *[T32AxCRB]* | 0x0084 |
| Capture Control Register B | *[T32AxCAPCRB]* | 0x0088 |
| Output Control Register B0 | *[T32AxOUTCRB0]* | 0x008C |
| Output Control Register B1 | *[T32AxOUTCRB1]* | 0x0090 |
| Status Register B | *[T32AxSTB]* | 0x0094 |
| Interrupt Mask Register B | *[T32AxIMB]* | 0x0098 |
| Counter Capture Register B | *[T32AxTMRB]* | 0x009C |
| Counter Reload Register B | *[T32AxRELDB]* | 0x00A0 |
| Timer Register B0 | *[T32AxRGB0]* | 0x00A4 |
| Timer Register B1 | *[T32AxRGB1]* | 0x00A8 |
| Capture Register B0 | *[T32AxCAPB0]* | 0x00AC |
| Capture Register B1 | *[T32AxCAPB1]* | 0x00B0 |
| DMA Request Enable Register B | *[T32AxDMAB]* | 0x00B4 |
| Compare Register B0 (Note) | *[T32AxCPB0]* | 0x00B8 |
| Compare Register B1 (Note) | *[T32AxCPB1]* | 0x00BC |

Note: When the base address is TYPE1, this register cannot be accessed.
For base address type, refer to "Product Information" of reference manual.

**(5) Timer C register (32-bit timer)**

| Register name | | Address(Base+) |
|---|---|---|
| RUN Register C | *[T32AxRUNC]* | 0x00C0 |
| Counter Control Register C | *[T32AxCRC]* | 0x00C4 |
| Capture Control Register C | *[T32AxCAPCRC]* | 0x00C8 |
| Output Control Register C0 | *[T32AxOUTCRC0]* | 0x00CC |
| Output Control Register C1 | *[T32AxOUTCRC1]* | 0x00D0 |
| Status Register C | *[T32AxSTC]* | 0x00D4 |
| Interrupt Mask Register C | *[T32AxIMC]* | 0x00D8 |
| Counter Capture Register C | *[T32AxTMRC]* | 0x00DC |
| Counter Reload Register C | *[T32AxRELDC]* | 0x00E0 |
| Timer Register C0 | *[T32AxRGC0]* | 0x00E4 |
| Timer Register C1 | *[T32AxRGC1]* | 0x00E8 |
| Capture Register C0 | *[T32AxCAPC0]* | 0x00EC |
| Capture Register C1 | *[T32AxCAPC1]* | 0x00F0 |
| DMA Request Enable Register C | *[T32AxDMAC]* | 0x00F4 |
| Pulse Count Control Register | *[T32AxPLSCR]* | 0x00F8 |
| Compare Register C0 (Note) | *[T32AxCPC0]* | 0x00FC |
| Compare Register C1 (Note) | *[T32AxCPC1]* | 0x0100 |

Note: When the base address is TYPE1, this register cannot be accessed.
For base address, refer to "Product Information" of reference manual.

## 4.2. Details of the Common Register

### 4.2.1. *[T32AxMOD]* (Mode Register)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:2 | - | 0 | R | Read as "0". |
| 1 | HALT | 0 | R/W | Sets the operation in HALT mode when debugging.<br>　0: Operation<br>　1: Stop<br><br>Sets the counter operation when the T32A enters HALT mode while debugging. |
| 0 | MODE32 | 0 | R/W | Sets the operation mode.<br>　0: 16-bit mode<br>　1: 32-bit mode<br><br>When this bit is set to "0", T32A operates at 16-bit mode that is enable timer A/ timer B and disable Timer C.<br>When this bit is set to "1", T32A operates at 32-bit mode that is enable timer C and disable Timer A/Timer B.<br>When the operation mode is changed, all registers of Timer A, timer B, and timer C are initialized except *[T32AxMOD]*.<br>When attempting to change the operation mode, check whether the timer is stopping using *[T32AxRUNA]* <RUNFLGA> and *[T32AxRUNB]*<RUNFLGB> (16-bit mode), or *[T32AxRUNC]* <RUNFLGC> (32-bit mode). |

Note: When *[T32AxRUNA]*<RUNA> = 0, *[T32AxRUNB]*<RUNB> = 0 and *[T32AxRUNC]*<RUNC> =0,
　　　 set the *[T32AxMOD]* register.

## 4.3. Details of Timer A Register

### 4.3.1. *[T32AxRUNA]* (Run Register A)

| Bit | Bit symbol | After reset | Type | Function |
|-----|-----------|-------------|------|----------|
| 31:5 | - | 0 | R | Read as "0". |
| 4 | RUNFLGA | 0 | R | Indicates the operation flag of Timer A.<br>　0: Stop<br>　1: Operation |
| 3 | - | 0 | R | Read as "0". |
| 2 | SFTSTPA | 0 | W | Stops the operation by program.<br>　0: -<br>　1: Stops the counter operation.<br><br>When this bit is set to "1" during the counter operation, the counter stops. Writing "0" has no meaning.<br>Read as "0". |
| 1 | SFTSTAA | 0 | W | Starts the operation by program.<br>　0: -<br>　1: Starts the counter operation.<br><br>When <RUNA> is set to "1", if <SFTSTAA> is set to "1", the counter starts operation. Writing "0" has no meaning.<br>Read as "0". |
| 0 | RUNA | 0 | R/W | Controls the operation of Timer A.<br>　0: Disabled<br>　1: Enabled<br><br>When <RUNA> is set to "1", Timer A enters the state where it waits for a startup factor. If a startup factor occurs, counting starts. Before <RUNA> is set to "1", check whether <RUNFLGA> is "0". |

### 4.3.2. *[T32AxCRA]* (Counter Control Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31 | - | 0 | R | Read as "0". |
| 30:28 | PRSCLA[2:0] | 000 | R/W | Selects the prescaler division ratio.<br>000: 1/1<br>001: 1/2<br>010: 1/8<br>011: 1/32<br>100: 1/128<br>101: 1/256<br>110: 1/512<br>111: 1/1024 |
| 27 | - | 0 | R | Read as "0". |
| 26:24 | CLKA[2:0] | 000 | R/W | Selects the count clock.<br>000: Prescaler output<br>001: Internal trigger (Note1)<br>010: On the rising edge of an output from other timer (T32AxTRGINAPHCK) (Note1)<br>011: On the falling edge of an output from other timer (T32AxTRGINAPHCK) (Note1)<br>100: On the rising edge of the external trigger (T32AxINA0).<br>101: On the falling edge of the external trigger (T32AxINA0).<br>110-111: Reserved |
| 23:21 | - | 0 | R | Read as "0". |
| 20 | WBFA | 0 | R/W | Controls double-buffering.<br>0: Disabled<br>1: Enabled |
| 19:18 | - | 0 | R | Read as "0". |
| 17:16 | UPDNA[1:0] | 00 | R/W | Selects the counter operation.<br>00: Up counting<br>01: Down counting<br>10: Up/down counting<br>11: Reserved<br><br>In case of Up/down counting, the counter starts counting up after the operation starts. Then, when the counter value matches Timer Register A1, the counter changes its operation to the down count operation.<br>Subsequently, when the counter value becomes "0x0000", the counter changes its operation to the up count operation. This operation is repeated. |
| 15:11 | - | 0 | R | Read as "0". |
| 10:8 | RELDA[2:0] | 000 | R/W | Counter reload condition<br>000: None (free running)<br>001: Internal trigger (Note1)<br>010: On the rising edge of the external trigger (T32AxINA0).<br>011: On the falling edge of the external trigger (T32AxINA0).<br>100: On the rising edge of an output from other timer (T32AxTRGINAPHCK) (Note1)<br>101: On the falling edge of an output from other timer (T32AxTRGINAPHCK) (Note1)<br>110: Synchronous operation (slave channel)<br>111: A match between the counter and Timer Register A1 |
| 7 | - | 0 | R | Read as "0". |

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 6:4 | STOPA[2:0] | 000 | R/W | Sets the counter stop condition.<br>  000: No trigger is used.<br>  001: Internal trigger (Note1)<br>  010: On the rising edge of the external trigger (T32AxINA0).<br>  011: On the falling edge of the external trigger (T32AxINA0).<br>  100: On the rising edge of an output from other timer<br>     (T32AxTRGINAPHCK) (Note1)<br>  101: On the falling edge of an output from other timer<br>     (T32AxTRGINAPHCK) (Note1)<br>  110: Synchronous operation (slave channel)<br>  111: A match between the counter and Timer Register A1 |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | STARTA[2:0] | 000 | R/W | Sets the counter start condition.<br>  000: No trigger is used.<br>  001: Internal trigger (Note1)<br>  010: On the rising edge of the external trigger (T32AxINA0).<br>  011: On the falling edge of the external trigger (T32AxINA0).<br>  100: On the rising edge of an output from other timer<br>     (T32AxTRGINAPHCK) (Note1)<br>  101: On the falling edge of an output from other timer<br>     (T32AxTRGINAPHCK) (Note1)<br>  110: Synchronous operation (slave channel)<br>  111: Reserved |

Note1: For details, refer to "Product Information" of reference manual.
Note2: Do not make any changes of *[T32AxCRA]* register while the *[T32AxRUNA]*<RUNA> = 1.

### 4.3.3. *[T32AxOUTCRA0]* (Output Control Register A0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:2 | - | 0 | R | Read as "0". |
| 1:0 | OCRA[1:0] | 00 | W | Controls T32AxOUTA.<br>  00: No change.<br>  01: Set ("High")<br>  10: Clear ("Low")<br>  11: Reversed<br><br>Read as "00". |

### 4.3.4. *[T32AxOUTCRA1]* (Output Control Register A1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7:6 | OCRCAPA1[1:0] | 00 | R/W | Controls T32AxOUTA by Capture Register A1<br>Controls T32AxOUTA when the counter value is captured in ***[T32AxCAPA1]***.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |
| 5:4 | OCRCAPA0[1:0] | 00 | R/W | Controls T32AxOUTA by Capture Register A0<br>Controls T32AxOUTA when the counter value is captured in ***[T32AxCAPA0]***.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |
| 3:2 | OCRCMPA1[1:0] | 00 | R/W | Controls T32AxOUTA by comparator A1<br>Controls T32AxOUTA when the counter value matches with ***[T32AxRGA1]***.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |
| 1:0 | OCRCMPA0[1:0] | 00 | R/W | Controls T32AxOUTA by comparator A0<br>Controls T32AxOUTA when the counter value matches with ***[T32AxRGA0]***.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |

Note: Do not make any changes of *[T32AxOUTCRA1]* register while the *[T32AxRUNA]*<RUNA> = 1.

### 4.3.5. [T32AxRGA0] (Timer Register A0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | RGA0[15:0] | 0x0000 | R/W | Sets the value for comparing with the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1 Note of Timer Register Setting when Outputting PPG".

### 4.3.6. *[T32AxRGA1]* (Timer Register A1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | RGA1[15:0] | 0x0000 | R/W | Sets the value for comparing with the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1 Note of Timer Register Setting when Outputting PPG".

### 4.3.7. *[T32AxTMRA]* (Counter Capture Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | TMRA[15:0] | 0x0000 | R | When this register is read during the counter operation, the current value of counter will be captured. And this counter value can be read.<br>When the counter is stopping, the last captured value can be read. |

### 4.3.8. *[T32AxRELDA]* (Counter Reload Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | RELDA[15:0] | 0x0000 | R/W | Sets the value to be reloaded to the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1 Note of Timer Register Setting when Outputting PPG".

### 4.3.9. *[T32AxCAPCRA]* (Capture Control Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:7 | - | 0 | R | Read as "0". |
| 6:4 | CAPMA1[2:0] | 000 | R/W | Sets the capture timing of *[T32AxCAPA1]*.<br>000: Disabled<br>001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>010: On the rising edge of the T32AxINA0 input pin<br>011: On the falling edge of the T32AxINA0 input pin.<br>100: On the rising edge of the T32AxINA1 input pin.<br>101: On the falling edge of the T32AxINA1 input pin.<br>110: On the rising edge of an output from other timer<br>　(T32AxTRGINAPHCK) (Note1)<br>111: On the falling edge of an output from other timer<br>　(T32AxTRGINAPHCK) (Note1) |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | CAPMA0[2:0] | 000 | R/W | Sets the capture timing of *[T32AxCAPA0]*.<br>000: Disabled<br>001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>010: On the rising edge of the T32AxINA0 input pin<br>011: On the falling edge of the T32AxINA0 input pin.<br>100: On the rising edge of the T32AxINA1 input pin.<br>101: On the falling edge of the T32AxINA1 input pin.<br>110: On the rising edge of an output from other timer<br>　(T32AxTRGINAPHCK) (Note1)<br>111: On the falling edge of an output from other timer<br>　(T32AxTRGINAPHCK) (Note1) |

Note1: For details, refer to "Product Information" of reference manual.
Note2: Do not make any changes of *[T32AxCAPCRA]* register while the *[T32AxRUNA]*<RUNA> = 1.

### 4.3.10. *[T32AxCAPA0]* (Capture Register A0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CAPA0[15:0] | 0x0000 | R | Captures the value in the counter. |

### 4.3.11. *[T32AxCAPA1]* (Capture Register A1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CAPA1[15:0] | 0x0000 | R | Captures the value in the counter. |

### 4.3.12. *[T32AxIMA]* (Interrupt Mask Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:4 | - | 0 | R | Read as "0". |
| 3 | IMUFA | 0 | R/W | Controls to mask the underflow interrupt request.<br>    0: An interrupt request is not masked.<br>    1: Masks the interrupt request.<br><br>Controls whether the underflow interrupt request is masked. |
| 2 | IMOFA | 0 | R/W | Controls to mask the overflow interrupt request.<br>    0: An interrupt request is not masked.<br>    1: Masks the interrupt request.<br><br>Controls whether the overflow interrupt request is masked. |
| 1 | IMA1 | 0 | R/W | Control to mask the match detection interrupt request (*[T32AxRGA1]*).<br>    0: An interrupt request is not masked.<br>    1: Masks the interrupt request.<br><br>Controls whether the match detection interrupt request is masked when the counter matches *[T32AxRGA1]*. |
| 0 | IMA0 | 0 | R/W | Control to mask the match detection interrupt request (*[T32AxRGA0]*).<br>    0: An interrupt request is not masked.<br>    1: Masks the interrupt request.<br><br>Controls whether the match detection interrupt request is masked when the counter matches *[T32AxRGA0]*. |

Note1: Even when the interrupt mask register (*[T32AxIMA]*) is enabled, a status flag of the counter is set to the *[T32AxSTA]*.

Note2: Do not make any changes of *[T32AxIMA]* register while the *[T32AxRUNA]*<RUNA> = 1.

### 4.3.13. *[T32AxSTA]* (Status Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:4 | - | 0 | R | Read as "0". |
| 3 | INTUFA | 0 | R | Indicates an underflow flag.<br>　0: No underflow occurred.<br>　1: An underflow occurred.<br><br>When the counter underflows, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 2 | INTOFA | 0 | R | Indicates an overflow flag.<br>　0: No overflow occurred.<br>　1: An overflow occurred.<br><br>When the counter overflows, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 1 | INTA1 | 0 | R | Indicates a match flag.<br>　0: No match is detected.<br>　1: A match between the counter and *[T32AxRGA1]*.<br><br>When a match between the counter value and the Timer Register A1 (*[T32AxRGA1]*) is detected, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 0 | INTA0 | 0 | R | Indicates a match flag.<br>　0: No match is detected.<br>　1: A match between the counter and *[T32AxRGA0]*<br><br>When a match between the counter value and the Timer Register A0 (*[T32AxRGA0]*) is detected, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |

Note1: Even when the Interrupt Mask Register (*[T32AxIMA]*) is enabled, a status flag is set to the *[T32AxSTA]* register.

Note2: Only the interrupt request that is not masked with *[T32AxIMA]* is output to the CPU.

Note3: When setting flag by internal signal and clearing flag by program occur at the same time, clearing flag has priority.

### 4.3.14. *[T32AxDMAA]* (DMA Request Enable Register A)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:3 | - | 0 | R | Read as "0". |
| 2 | DMAENA2 | 0 | R/W | Selects a DMA request occurrence condition: A match between the counter and the Timer Register A1 (*[T32AxRGA1]*)<br>0: Disabled<br>1: Enabled |
| 1 | DMAENA1 | 0 | R/W | Selects a DMA request occurrence condition: Input capture 1 (*[T32AxCAPA1]*)<br>0: Disabled<br>1: Enabled |
| 0 | DMAENA0 | 0 | R/W | Selects a DMA request occurrence condition: Input capture 0 (*[T32AxCAPA0]*)<br>0: Disabled<br>1: Enabled |

Note: Do not make any changes of *[T32AxDMAA]* register while the *[T32AxRUNA]*<RUNA> = 1.

### 4.3.15. *[T32AxCPA0]* (Compare Register A0)

| Bit | Bit symbol | After reset | Type | Function |
|------|------------|-------------|------|----------|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CRGA0[15:0] | 0x0000 | R | Read the value for comparing with the counter. |

Note: When the base address is TYPE1, this register cannot be accessed. For base address, refer to "Product Information" of reference manual.

### 4.3.16. *[T32AxCPA1]* (Compare Register A1)

| Bit | Bit symbol | After reset | Type | Function |
|------|------------|-------------|------|----------|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CRGA1[15:0] | 0x0000 | R | Read the value for comparing with the counter. |

Note: When the base address is TYPE1, this register cannot be accessed. For base address, refer to "Product Information" of reference manual.

## 4.4. Details of Timer B Register

### 4.4.1. *[T32AxRUNB]* (Run Register B)

| Bit | Bit symbol | After reset | Type | Function |
|------|------------|-------------|------|----------|
| 31:5 | - | 0 | R | Read as "0". |
| 4 | RUNFLGB | 0 | R | Indicates an operation flag of Timer B.<br>　0: Stop<br>　1: Operation |
| 3 | - | 0 | R | Read as "0". |
| 2 | SFTSTPB | 0 | W | Stops the counter operation by program.<br>　0: -<br>　1: Stops the counter operation.<br><br>When this bit is set to "1" during the counter operation, the counter operation stops. Writing "0" has no meaning.<br>Read as "0". |
| 1 | SFTSTAB | 0 | W | Starts the counter operation by program.<br>　0: -<br>　1: Starts the counter operation.<br><br>When <RUNB> is set to "1", if <SFTSTAB> is set to "1", the counter starts operation. Writing "0" has no meaning.<br>Read as "0". |
| 0 | RUNB | 0 | R/W | Controls the operation of Timer B.<br>　0: Disabled<br>　1: Enabled<br><br>When <RUNB> is set to "1", Timer B enters the state where it waits for a startup factor. If a startup factor occurs, counting starts. Before <RUNB> is set to "1", check whether <RUNFLGB> is "0". |

## 4.4.2. *[T32AxCRB]* (Counter Control Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31 | - | 0 | R | Read as "0". |
| 30:28 | PRSCLB[2:0] | 000 | R/W | Selects the prescaler division ratio.<br>  000: 1/1<br>  001: 1/2<br>  010: 1/8<br>  011: 1/32<br>  100: 1/128<br>  101: 1/256<br>  110: 1/512<br>  111: 1/1024 |
| 27 | - | 0 | R | Read as "0". |
| 26:24 | CLKB[2:0] | 000 | R/W | Selects the count clock.<br>  000: Prescaler output<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of an output from other timer<br>       (T32AxTRGINBPHCK) (Note1)<br>  011: On the falling edge of an output from other timer<br>       (T32AxTRGINBPHCK) (Note1)<br>  100: On the rising edge of the external trigger (T32AxINB0).<br>  101: On the falling edge of the external trigger (T32AxINB0).<br>  110-111: Reserved |
| 23:21 | - | 0 | R | Read as "0". |
| 20 | WBFB | 0 | R/W | Controls double-buffering.<br>  0: Disabled<br>  1: Enabled |
| 19:18 | - | 0 | R | Read as "0". |
| 17:16 | UPDNB[1:0] | 00 | R/W | Selects the counter operation.<br>  00: Up counting<br>  01: Down counting<br>  10: Up/down counting<br>  11: Reserved<br>In case of Up/down counting, the counter starts counting up after the operation starts. Then, when the counter value matches Timer Register B1, the counter changes its operation to the down count operation.<br>Subsequently, when the counter value becomes "0x0000", the counter changes its operation to the up count operation. This operation is repeated. |
| 15:11 | - | 0 | R | Read as "0". |
| 10:8 | RELDB[2:0] | 000 | R/W | Counter reload condition<br>  000: None (free running)<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of the external trigger (T32AxINB0).<br>  011: On the falling edge of the external trigger (T32AxINB0).<br>  100: On the rising edge of an output from other timer<br>       (T32AxTRGINBPHCK) (Note1)<br>  101: On the falling edge of an output from other timer<br>       (T32AxTRGINBPHCK) (Note1)<br>  110: Synchronous operation (slave channel)<br>  111: A match between the counter and Timer Register B1 |
| 7 | - | 0 | R | Read as "0". |

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 6:4 | STOPB[2:0] | 000 | R/W | Sets the counter stop condition.<br>  000: No trigger is used.<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of the external trigger (T32AxINB0).<br>  011: On the falling edge of the external trigger (T32AxINB0).<br>  100: On the rising edge of an output from other timer<br>    (T32AxTRGINBPHCK) (Note1)<br>  101: On the falling edge of an output from other timer<br>    (T32AxTRGINBPHCK) (Note1)<br>  110: Synchronous operation (slave channel)<br>  111: A match between the counter and Timer Register B1 |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | STARTB[2:0] | 000 | R/W | Sets the counter start condition.<br>  000: No trigger is used.<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of the external trigger (T32AxINB0).<br>  011: On the falling edge of the external trigger (T32AxINB0).<br>  100: On the rising edge of an output from other timer<br>    (T32AxTRGINBPHCK) (Note1)<br>  101: On the falling edge of an output from other timer<br>    (T32AxTRGINBPHCK) (Note1)<br>  110: Synchronous operation (slave channel)<br>  111: Reserved |

Note1: For details, refer to "Product Information" of reference manual.
Note2: Do not make any changes of *[T32AxCRB]* register while the *[T32AxRUNB]*<RUNB> = 1.

### 4.4.3. *[T32AxOUTCRB0]* (Output Control Register B0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:2 | - | 0 | R | Read as "0". |
| 1:0 | OCRB[1:0] | 00 | W | Controls T32AxOUTB.<br>  00: No change.<br>  01: Set ("High")<br>  10: Clear ("Low")<br>  11: Reversed<br><br>Read as "00". |

### 4.4.4. *[T32AxOUTCRB1]* (Output Control Register B1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7:6 | OCRCAPB1[1:0] | 00 | R/W | Controls T32AxOUTB by Capture Register B1<br>Controls T32AxOUTB when the counter value is captured in *[T32AxCAPB1]*.<br>  00: Invalid<br>  01: Set ("High")<br>  10: Clear ("Low")<br>  11: Reversed |
| 5:4 | OCRCAPB0[1:0] | 00 | R/W | Controls T32AxOUTB by Capture Register B0<br>Controls T32AxOUTB when the counter value is captured in *[T32AxCAPB0]*.<br>  00: Invalid<br>  01: Set ("High")<br>  10: Clear ("Low")<br>  11: Reversed |

| | | | | |
|---|---|---|---|---|
| 3:2 | OCRCMPB1[1:0] | 00 | R/W | Controls T32AxOUTB by comparator B1<br>Controls T32AxOUTB when the counter value matches with **[T32AxRGB1]**.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |
| 1:0 | OCRCMPB0[1:0] | 00 | R/W | Controls T32AxOUTB by comparator B0<br>Controls T32AxOUTB when the counter value matches with **[T32AxRGB0]**.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |

Note: Do not make any changes of *[T32AxOUTCRB1]* register while the *[T32AxRUNB]*<RUNB> = 1.

### 4.4.5. *[T32AxRGB0]* (Timer Register B0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | RGB0[15:0] | 0x0000 | R/W | Sets the value for comparing with the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1 Note of Timer Register Setting when Outputting PPG".

### 4.4.6. *[T32AxRGB1]* (Timer Register B1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | RGB1[15:0] | 0x0000 | R/W | Sets the value for comparing with the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1 Note of Timer Register Setting when Outputting PPG".

### 4.4.7. *[T32AxTMRB]* (Counter Capture Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | TMRB[15:0] | 0x0000 | R | When this register is read during the counter operation, the current value of counter will be captured. And this counter value can be read.<br>When the counter is stopping, the last captured value can be read. |

### 4.4.8. *[T32AxRELDB]* (Counter Reload Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | RELDB[15:0] | 0x0000 | R/W | Sets the value to be reloaded to the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1 Note of Timer Register Setting when Outputting PPG".

### 4.4.9. *[T32AxCAPCRB]* (Capture Control Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:7 | - | 0 | R | Read as "0". |
| 6:4 | CAPMB1[2:0] | 000 | R/W | Sets the capture timing of *[T32AxCAPB1]*.<br>  000: Disabled<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of the T32AxINB0 input pin<br>  011: On the falling edge of the T32AxINB0 input pin.<br>  100: On the rising edge of the T32AxINB1 input pin.<br>  101: On the falling edge of the T32AxINB1 input pin.<br>  110: On the rising edge of an output from other timer<br>     (T32AxTRGINBPHCK) (Note1)<br>  111: On the falling edge of an output from other timer<br>     (T32AxTRGINBPHCK) (Note1) |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | CAPMB0[2:0] | 000 | R/W | Sets the capture timing of *[T32AxCAPB0]*.<br>  000: Disabled<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of the T32AxINB0 input pin<br>  011: On the falling edge of the T32AxINB0 input pin.<br>  100: On the rising edge of the T32AxINB1 input pin.<br>  101: On the falling edge of the T32AxINB1 input pin.<br>  110: On the rising edge of an output from other timer<br>     (T32AxTRGINBPHCK) (Note1)<br>  111: On the falling edge of an output from other timer<br>     (T32AxTRGINBPHCK) (Note1) |

Note1: For details, refer to "Product Information" of reference manual.
Note2: Do not make any changes of *[T32AxCAPCRB]* register while the *[T32AxRUNB]*<RUNB> = 1.

### 4.4.10. *[T32AxCAPB0]* (Capture Register B0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CAPB0[15:0] | 0x0000 | R | Captures the value in the counter. |

### 4.4.11. *[T32AxCAPB1]* (Capture Register B1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CAPB1[15:0] | 0x0000 | R | Captures the value in the counter. |

## 4.4.12. *[T32AxIMB]* (Interrupt Mask Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:4 | - | 0 | R | Read as "0". |
| 3 | IMUFB | 0 | R/W | Controls to mask the underflow interrupt request.<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the underflow interrupt request is masked. |
| 2 | IMOFB | 0 | R/W | Controls to mask the overflow interrupt request.<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the overflow interrupt request is masked. |
| 1 | IMB1 | 0 | R/W | Control to mask the match detection interrupt request (*[T32AxRGB1]*).<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the match detection interrupt request is masked when the counter matches *[T32AxRGB1]*. |
| 0 | IMB0 | 0 | R/W | Control to mask the match detection interrupt request (*[T32AxRGB0]*).<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the match detection interrupt request is masked when the counter matches *[T32AxRGB0]*. |

Note1: Even when the interrupt mask register (*[T32AxIMB]*) is enabled, a status flag of the counter is set to the *[T32AxSTB]*.

Note2: Do not make any changes of *[T32AxIMB]* register while the *[T32AxRUNB]*<RUNB> = 1.

### 4.4.13. *[T32AxSTB]* (Status Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:4 | - | 0 | R | Read as "0". |
| 3 | INTUFB | 0 | R | Indicates an underflow flag.<br>  0: No underflow occurred.<br>  1: An underflow occurred.<br><br>When the counter underflows, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 2 | INTOFB | 0 | R | Indicates an overflow flag.<br>  0: No overflow occurred.<br>  1: An overflow occurred.<br><br>When the counter overflows, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 1 | INTB1 | 0 | R | Indicates a match flag.<br>  0: No match is detected.<br>  1: A match between the counter and *[T32AxRGB1]* is detected.<br><br>When a match between the counter value and the Timer Register B1 (*[T32AxRGB1]*) is detected, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 0 | INTB0 | 0 | R | Indicates a match flag.<br>  0: No match is detected.<br>  1: A match between the counter and *[T32AxRGB0]* is detected.<br><br>When a match between the counter value and the Timer Register B0 (*[T32AxRGB0]*) is detected, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |

Note1: Even when the Interrupt Mask Register *[T32AxIMB]* is enabled, a status flag is set to the *[T32AxSTB]* register.

Note2: Only the interrupt request that is not masked with *[T32AxIMB]* is output to the CPU.

Note3: When setting flag by internal signal and clearing flag by program occur at the same time, clearing flag has priority.

### 4.4.14. *[T32AxDMAB]* (DMA Request Enable Register B)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:3 | - | 0 | R | Read as "0". |
| 2 | DMAENB2 | 0 | R/W | Selects a DMA request occurrence condition: A match between the counter and the Timer Register B1 (*[T32AxRGB1]*)<br>  0: Disabled<br>  1: Enabled |
| 1 | DMAENB1 | 0 | R/W | Selects a DMA request occurrence condition: Input capture 1 (*[T32AxCAPB1]*)<br>  0: Disabled<br>  1: Enabled |
| 0 | DMAENB0 | 0 | R/W | Selects a DMA request occurrence condition: Input capture 0 (*[T32AxCAPB0]*)<br>  0: Disabled<br>  1: Enabled |

Note: Do not make any changes of *[T32AxDMAB]* register while the *[T32AxRUNB]*<RUNB> = 1.

### 4.4.15. *[T32AxCPB0]* (Compare Register B0)

| Bit | Bit symbol | After reset | Type | Function |
|-----|-----------|-------------|------|----------|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CRGB0[15:0] | 0x0000 | R | Read the value for comparing with the counter. |

Note: When the base address is TYPE1, this register cannot be accessed. For base address, refer to "Product Information" of reference manual.

### 4.4.16. *[T32AxCPB1]* (Compare Register B1)

| Bit | Bit symbol | After reset | Type | Function |
|-----|-----------|-------------|------|----------|
| 31:16 | - | 0 | R | Read as "0". |
| 15:0 | CRGB1[15:0] | 0x0000 | R | Read the value for comparing with the counter. |

Note: When the base address is TYPE1, this register cannot be accessed. For base address, refer to "Product Information" of reference manual.

## 4.5. Details of Timer C Register

### 4.5.1. *[T32AxRUNC]* (Run Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:5 | - | 0 | R | Read as "0". |
| 4 | RUNFLGC | 0 | R | Indicates an operation flag of Timer C.<br>　0: Stop<br>　1: Operation |
| 3 | - | 0 | R | Read as "0". |
| 2 | SFTSTPC | 0 | W | Stops the operation by program.<br>　0: -<br>　1: Stops the counter operation.<br><br>When this bit is set to "1" during the counter operation, the counter stops. Writing "0" has no meaning.<br>Read as "0". |
| 1 | SFTSTAC | 0 | W | Starts the operation by program.<br>　0: -<br>　1: Starts the counter operation.<br><br>When <RUNC> is set to "1", if <SFTSTAC> is set to "1", the counter starts operation. Writing "0" has no meaning.<br>Read as "0". |
| 0 | RUNC | 0 | R/W | Controls the operation of Timer C.<br>　0: Disabled<br>　1: Enabled<br><br>When <RUNC> is set to "1", Timer A enters the state where it waits for a startup factor. If a startup factor occurs, counting starts. Before <RUNC> is set to "1", check whether <RUNFLGC> is "0". |

## 4.5.2. *[T32AxCRC]* (Counter Control Register C)

| Bit | Bit symbol | After reset | Type | Function |
|-----|-----------|-------------|------|----------|
| 31 | - | 0 | R | Read as "0". |
| 30:28 | PRSCLC[2:0] | 000 | R/W | Selects the prescaler division ratio.<br>  000: 1/1<br>  001: 1/2<br>  010: 1/8<br>  011: 1/32<br>  100: 1/128<br>  101: 1/256<br>  110: 1/512<br>  111: 1/1024 |
| 27 | - | 0 | R | Read as "0". |
| 26:24 | CLKC[2:0] | 000 | R/W | Selects the count clock.<br>  000: Prescaler output<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of an output from other timer<br>      (T32AxTRGINCPHCK) (Note1)<br>  011: On the falling edge of an output from other timer<br>      (T32AxTRGINCPHCK) (Note1)<br>  100: On the rising edge of the external trigger (T32AxINC0).<br>  101: On the falling edge of the external trigger (T32AxINC0).<br>  110-111: Reserved |
| 23:21 | - | 0 | R | Read as "0". |
| 20 | WBFC | 0 | R/W | Controls double-buffering.<br>  0: Disabled<br>  1: Enabled |
| 19:18 | - | 0 | R | Read as "0". |
| 17:16 | UPDNC[1:0] | 00 | R/W | Selects the counter operation.<br>  00: Up counting<br>  01: Down counting<br>  10: Up/down counting<br>  11: Pulse counting For details, refer to the pulse count control register (*[T32AxPLSCR]*).<br><br>In case of Up/down counting, the counter starts counting up after the operation starts. Then, when the counter value matches Timer Register C1, the counter changes its operation to the down count operation.<br>Subsequently, when the counter value becomes "0x0000", the counter changes its operation to the up count operation. This operation is repeated. |
| 15:11 | - | 0 | R | Read as "0". |
| 10:8 | RELDC[2:0] | 000 | R/W | Counter reload condition<br>  000: None (free running)<br>  001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>  010: On the rising edge of the external trigger (T32AxINC0).<br>  011: On the falling edge of the external trigger (T32AxINC0).<br>  100: On the rising edge of an output from other timer<br>      (T32AxTRGINCPHCK) (Note1)<br>  101: On the falling edge of an output from other timer<br>      (T32AxTRGINCPHCK) (Note1)<br>  110: Synchronous operation (slave channel)<br>  111: A match between the counter and Timer Register C1 |
| 7 | - | 0 | R | Read as "0". |

| | | | | |
|---|---|---|---|---|
| 6:4 | STOPC[2:0] | 000 | R/W | Sets the counter stop condition.<br>　000: No trigger is used.<br>　001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>　010: On the rising edge of the external trigger (T32AxINC0).<br>　011: On the falling edge of the external trigger (T32AxINC0).<br>　100: On the rising edge of an output from other timer<br>　　　(T32AxTRGINCPHCK) (Note1)<br>　101: On the falling edge of an output from other timer<br>　　　(T32AxTRGINCPHCK) (Note1)<br>　110: Synchronous operation (slave channel)<br>　111: A match between the counter and Timer Register C1 |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | STARTC[2:0] | 000 | R/W | Sets the counter start condition.<br>　000: No trigger is used.<br>　001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>　010: On the rising edge of the external trigger (T32AxINC0).<br>　011: On the falling edge of the external trigger (T32AxINC0).<br>　100: On the rising edge of an output from other timer<br>　　　(T32AxTRGINCPHCK) (Note1)<br>　101: On the falling edge of an output from other timer<br>　　　(T32AxTRGINCPHCK) (Note1)<br>　110: Synchronous operation (slave channel)<br>　111: Reserved |

Note1: For details, refer to "Product Information" of reference manual.
Note2: Do not make any changes of *[T32AxCRC]* register while the *[T32AxRUNC]*<RUNC> = 1.

### 4.5.3. *[T32AxOUTCRC0]* (Output Control Register C0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:2 | - | 0 | R | Read as "0". |
| 1:0 | OCRC[1:0] | 00 | W | Controls T32AxOUTC.<br>　00: No change.<br>　01: Set ("High")<br>　10: Clear ("Low")<br>　11: Reversed<br><br>Read as "00". |

### 4.5.4. *[T32AxOUTCRC1]* (Output Control Register C1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:8 | - | 0 | R | Read as "0". |
| 7:6 | OCRCAPC1[1:0] | 00 | R/W | Controls T32AxOUTC by Capture Register C1<br>Controls T32AxOUTC when the counter value is captured in *[T32AxCAPC1]*.<br>　00: Invalid<br>　01: Set ("High")<br>　10: Clear ("Low")<br>　11: Reversed |
| 5:4 | OCRCAPC0[1:0] | 00 | R/W | Controls T32AxOUTC by Capture Register C0<br>Controls T32AxOUTC when the counter value is captured in *[T32AxCAPC0]*.<br>　00: Invalid<br>　01: Set ("High")<br>　10: Clear ("Low")<br>　11: Reversed |

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 3:2 | OCRCMPC1[1:0] | 00 | R/W | Controls T32AxOUTC by comparator C1<br>Controls T32AxOUTC when the counter value matches with **[T32AxRGC1]**.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |
| 1:0 | OCRCMPC0[1:0] | 00 | R/W | Controls T32AxOUTC by comparator C0<br>Controls T32AxOUTC when the counter value matches with **[T32AxRGC0]**.<br>00: Invalid<br>01: Set ("High")<br>10: Clear ("Low")<br>11: Reversed |

Note: Do not make any changes of **[T32AxOUTCRC1]** register while the **[T32AxRUNC]**<RUNC> = 1.

### 4.5.5. *[T32AxRGC0]* (Timer Register C0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | RGC0[31:0] | 0x00000000 | R/W | Sets the value for comparing with the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1   Note of Timer Register Setting when Outputting PPG".

### 4.5.6. *[T32AxRGC1]* (Timer Register C1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | RGC1[31:0] | 0x00000000 | R/W | Sets the value for comparing with the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1   Note of Timer Register Setting when Outputting PPG".

### 4.5.7. *[T32AxTMRC]* (Counter Capture Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | TMRC[31:0] | 0x00000000 | R | When this register is read during the counter operation, the current value of counter will be captured. And this counter value can be read.<br>When the counter is stopping, the last captured value can be read. |

### 4.5.8. *[T32AxRELDC]* (Counter Reload Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | RELDC[31:0] | 0x00000000 | R/W | Sets the value to be reloaded to the counter. |

Note: About value of counter when PPG is output, refer to "Table 3.1   Note of Timer Register Setting when Outputting PPG".

### 4.5.9. *[T32AxCAPCRC]* (Capture Control Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:7 | - | 0 | R | Read as "0". |
| 6:4 | CAPMC1[2:0] | 000 | R/W | Sets the capture timing of *[T32AxCAPC1]*.<br>000: Disabled<br>001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>010: On the rising edge of the T32AxINC0 input pin<br>011: On the falling edge of the T32AxINC0 input pin.<br>100: On the rising edge of the T32AxINC1 input pin.<br>101: On the falling edge of the T32AxINC1 input pin.<br>110: On the rising edge of an output from other timer (T32AxTRGINCPHCK) (Note1)<br>111: On the falling edge of an output from other timer (T32AxTRGINCPHCK) (Note1) |
| 3 | - | 0 | R | Read as "0". |
| 2:0 | CAPMC0[2:0] | 000 | R/W | Sets the capture timing of *[T32AxCAPC0]*.<br>000: Disabled<br>001: Internal trigger (T32AxTRGINAPCK) (Note1)<br>010: On the rising edge of the T32AxINC0 input pin<br>011: On the falling edge of the T32AxINC0 input pin.<br>100: On the rising edge of the T32AxINC1 input pin.<br>101: On the falling edge of the T32AxINC1 input pin.<br>110: On the rising edge of an output from other timer (T32AxTRGINCPHCK) (Note1)<br>111: On the falling edge of an output from other timer (T32AxTRGINCPHCK) (Note1) |

Note1: For details, refer to "Product Information" of reference manual.
Note2: Do not make any changes of *[T32AxCAPCRC]* register while the *[T32AxRUNC]*<RUNC> = 1.

### 4.5.10. *[T32AxCAPC0]* (Capture Register C0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | CAPC0[31:0] | 0x00000000 | R | Captures the value in the counter. |

### 4.5.11. *[T32AxCAPC1]* (Capture Register C1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | CAPC1[31:0] | 0x00000000 | R | Captures the value in the counter. |

### 4.5.12. *[T32AxIMC]* (Interrupt Mask Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:5 | - | 0 | R | Read as "0". |
| 4 | IMSTERR | 0 | R/W | Controls to mask the status transition error interrupt request (only in 2-phase pulse count mode)<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Control whether the status transition error interrupt request is masked in 2-phase pulse count mode. |
| 3 | IMUFC | 0 | R/W | Controls to mask the underflow interrupt request.<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the underflow interrupt request is masked. |
| 2 | IMOFC | 0 | R/W | Controls to mask the overflow interrupt request.<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the overflow interrupt request is masked. |
| 1 | IMC1 | 0 | R/W | Control to mask the match detection interrupt request (*[T32AxRGC1]*).<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the match detection interrupt request is masked when the counter matches *[T32AxRGC1]*. |
| 0 | IMC0 | 0 | R/W | Control to mask the match detection interrupt request (*[T32AxRGC0]*).<br>  0: An interrupt request is not masked.<br>  1: Masks the interrupt request.<br><br>Controls whether the match detection interrupt request is masked when the counter matches *[T32AxRGC0]*. |

Note1: Even when the interrupt mask register (*[T32AxIMC]*) is enabled, a status flag of the counter is set to the *[T32AxSTC]*.

Note2: Do not make any changes of *[T32AxIMC]* register while the *[T32AxRUNC]*<RUNC> = 1.

## 4.5.13. *[T32AxSTC]* (Status Register C)

| Bit | Bit symbol | After reset | Type | Function |
|-----|-----------|-------------|------|----------|
| 31:5 | - | 0 | R | Read as "0". |
| 4 | INTSTERR | 0 | R | Indicates the status transition error interrupt request (only in 2-phase pulse count mode)<br>　0: No status transition error occurred.<br>　1: A status transition error occurred.<br><br>When a status transition error occurs in 2-phase pulse count mode, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 3 | INTUFC | 0 | R | Indicates an underflow flag.<br>　0: No underflow occurred.<br>　1: An underflow occurred.<br><br>When the counter underflows, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 2 | INTOFC | 0 | R | Indicates an overflow flag.<br>　0: No overflow occurred.<br>　1: An overflow occurred.<br><br>When the counter overflows, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 1 | INTC1 | 0 | R | Indicates a match flag.<br>　0: No match is detected.<br>　1: A match between the counter and *[T32AxRGC1]* is detected.<br><br>When a match between the counter value and the Timer Register C1 (*[T32AxRGC1]*) is detected, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |
| 0 | INTC0 | 0 | R | Indicates a match flag.<br>　0: No match is detected.<br>　1: A match between the counter and *[T32AxRGC0]* is detected.<br><br>When a match between the counter value and the Timer Register C0 (*[T32AxRGC0]*) is detected, this bit is set to "1". |
| | | | W | 0: Don't care.<br>1: Cleared to "0". |

Note1: Even when the Interrupt Mask Register *[T32AxIMC]* is enabled, a status flag is set to the *[T32AxSTC]* register.

Note2: Only the interrupt request that is not masked with *[T32AxIMC]* is output to the CPU.

Note3: When setting flag by internal signal and clearing flag by program occur at the same time, clearing flag has priority.

## 4.5.14. *[T32AxDMAC]* (DMA Request Enable Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:3 | - | 0 | R | Read as "0". |
| 2 | DMAENC2 | 0 | R/W | Selects a DMA request occurrence condition: A match between the counter and the Timer Register C1 (*[T32AxRGC1]*)<br>　0: Disabled<br>　1: Enabled |
| 1 | DMAENC1 | 0 | R/W | Selects a DMA request occurrence condition: Input capture 1 (*[T32AxCAPC1]*)<br>　0: Disabled<br>　1: Enabled |
| 0 | DMAENC0 | 0 | R/W | Selects a DMA request occurrence condition: Input capture 0 (*[T32AxCAPC0]*)<br>　0: Disabled<br>　1: Enabled |

Note: Do not make any changes of *[T32AxDMAC]* register while the *[T32AxRUNC]*<RUNC> = 1.

## 4.5.15. *[T32AxPLSCR]* (Pulse Counter control Register C)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:15 | - | 0 | R | Read as "0". |
| 14:12 | PDN[2:0] | 000 | R/W | Count down condition in 1-phase pulse counter mode.<br>　000: No count down.<br>　001: No count down.<br>　010: On the rising edge of T32AxINC0.<br>　011: On the falling edge of T32AxINC0.<br>　100: On the rising edge of T32AxINC1.<br>　101: On the falling edge of T32AxINC1.<br>　110: On the rising/falling edge of T32AxINC0.<br>　111: On the rising/falling edge of T32AxINC1. |
| 11 | - | 0 | R | Read as "0". |
| 10:8 | PUP[2:0] | 000 | R/W | Count up condition in 1-phase pulse count mode.<br>　000: No count up.<br>　001: No count up.<br>　010: On the rising edge of T32AxINC0<br>　011: On the falling edge of T32AxINC0<br>　100: On the rising edge of T32AxINC1<br>　101: On the falling edge of T32AxINC1<br>　110: On the rising edge/falling edge of T32AxINC0<br>　111: On the rising edge/falling edge of T32AxINC1 |
| 7:6 | - | 0 | R | Read as "0". |
| 5:4 | NF[1:0] | 00 | R/W | Selects the noise elimination time for T32AxINC0/T32AxINC1<br>For detail, refer to "Table 4.1　Noise Elimination <NF>".<br>　00: None<br>　01: Eliminates a signal less than $2/\Phi T0$ as noise.<br>　10: Eliminates a signal less than $4/\Phi T0$ as noise.<br>　11: Eliminates a signal less than $8/\Phi T0$ as noise. |
| 3:2 | - | 0 | R | Read as "0". |
| 1 | PDIR | 0 | R/W | Selects the direction of the 2-phase pulse counter.<br>　0: Positive direction<br>　1: Negative direction |
| 0 | PMODE | 0 | R/W | Selects the pulse counter mode.<br>　0: 2-phase pulse counter mode<br>　1: 1-phase pulse counter mode |

For details of noise elimination <NF>, see below:

**Table 4.1   Noise Elimination <NF>**

| <NF> | Level at which noise is surely eliminated. | Level at which signal is surely recognized | Level at which noise is surely eliminated (time) [CGSYSCR]<PRCKST[3:0]> = 0000 fc=40MHz |
|---|---|---|---|
| 01 | Less than 2/ΦT0 | 3/ΦT0 or more | 50ns |
| 10 | Less than 4/ΦT0 | 5/ΦT0 or more | 100ns |
| 11 | Less than 8/ΦT0 | 9/ΦT0 or more | 200ns |

## 4.5.16. *[T32AxCPC0]* (Compare Register C0)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | CRGC0[31:0] | 0x00000000 | R | Read the value for comparing with the counter. |

Note: When the base address is TYPE1, this register cannot be accessed. For base address, refer to "Product Information" of reference manual.

## 4.5.17. *[T32AxCPC1]* (Compare Register C1)

| Bit | Bit symbol | After reset | Type | Function |
|---|---|---|---|---|
| 31:0 | CRGC1[31:0] | 0x00000000 | R | Read the value for comparing with the counter. |

Note: When the base address is TYPE1, this register cannot be accessed. For base address, refer to "Product Information" of reference manual.

# 5. Use Cases

## 5.1. Interval Timer

T32A can generate interrupt to CPU periodically.
The following is the definition of the period.

| | Period |
|---|---|
| Up counting | *[T32AxRGA1]* – *[T32AxRELDA]* |
| Down counting | *[T32AxRELDA]* – *[T32AxRGA1]* |

The following case is an example when periodical interrupts are generated by counting up Timer A. Set the Reload Register *[T32AxRELDA]* to "0x0000" and set the interval time to the Timer Register *[T32AxRGA1]*, and set the counter reloaded condition. The reload condition is a compare match between the counter and *[T32AxRGA1]*. After the operation is started, an INTT32Ax interrupt is generated after the interval time has elapsed. The value of Reload Register is reloaded to the counter and continues counting up.

Counter Control Register A (*[T32AxCRA]*)
    Clock selection <CLKA[2:0]>        Any clock
    Count Operation <UPDNA[1:0]>     Up counting
    Start operation <STARTA[2:0]>     Any timing
    Stop operation <STOPA[2:0]>      Any timing
    Reload condition <RELDA[2:0]>     A match with *[T32AxRGA1]*



**Figure 5.1　Interval Timer (Count Up)**

The following case is an example when periodical interrupts are generated by counting down Timer A. Set the interval time to the Reload Register *[T32AxRELDA]*, and set "0x0000" to the Timer Register *[T32AxRGA1]*, and set the reload condition. The reload condition is a compare match between the counter and *[T32AxRGA1]*. After the operation is started, the counter counts down from the reloaded value to "0x0000". When the counter value reaches "0x0000", an INTT32AxA interrupt is generated. The counter is reloaded and continues counting down.

Counter Control Register A (*[T32AxCRA]*)
    Clock selection <CLKA[2:0]>        Any clock
    Count operation <UPDNA[1:0]>     Down counting
    Start operation <STARTA[2:0]>     Any timing
    Stop operation <STOPA[2:0]>      Any timing
    Reload Condition <RELDA[2:0]>     A match with *[T32AxRGA1]*
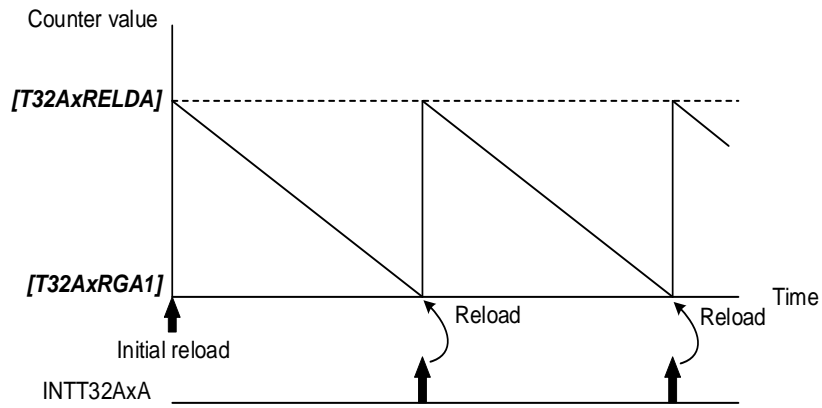
**Figure 5.2   Interval Timer (Count Down)**

## 5.2. Event Counter

An external trigger, internal trigger, or timer output from other timer can be selected as the event counter. As the count operation, up counting should be selected. The counter counts up on the rising edge/falling edge of the external trigger, on the rising edge/falling edge of the output from other timer, or at the occurrence of the internal trigger. The number of occurrences of the events can be read from *[T32AxTMRA]* as the count value.

Counter Control Register A (*[T32AxCRA]*)
| | |
|---|---|
| Clock selection <CLKA[2:0]> | External triggers / Internal triggers/ outputs from other timer. |
| Count operation <UPDNA[1:0]> | Up counting |
| Start operation <STARTA[2:0]> | Any timing |
| Stop operation <STOPA[2:0]> | Any timing |
| Reload condition <RELDA[2:0]> | Any condition |
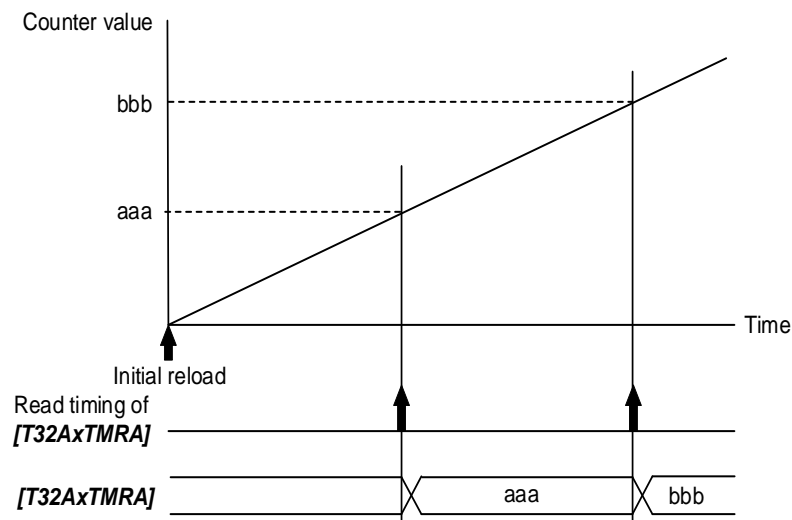


**Figure 5.3    Event Counter**

## 5.3. Programmable Rectangular Wave Output (PPG)

The T32A can output rectangular wave at any frequency and any duty ratio.

The following is the definition of the period and duty ratio.

|  | Period | Duty ratio |
|---|---|---|
| Up counting | *[T32AxRGA1]* – *[T32AxRELDA]* | *[T32AxRGA0]* – *[T32AxRELDA]* |

An output from Timer A (T32AxOUTA) is initially a "Low" signal. The level of the signal can be changed by *[T32AxOUTCRA0]*<OCRA>.

A timer output can be set, cleared, or reversed when Timer A counter matches the Timer Register (*[T32AxRGA0]*, *[T32AxRGA1]*). Any rectangular wave can be output from the T32AxOUTA pin.

When double buffering is enabled, a value in the register buffer A0/A1 is transferred to *[T32AxRGA0/A1]* when the counter matches *[T32AxRGA1]*. This can change the frequency and duty ratio without regard to the update timing of the timer register.

Counter Control Register A (*[T32AxCRA]*)
    Clock selection <CLKA[2:0]>         Any clock
    Count operation <UPDNA[1:0]>      Up counting
    Start operation <STARTA[2:0]>     Any timing
    Stop operation <STOPA[2:0]>      Any timing
    Reload condition <RELDA[2:0]>     A match with the Timer Register A1 (*[TX32ARGA1]*)
                                     (Reloaded "0x0000")
    Controls double-buffering <WBFA>   Enable

This figure shows a waveform when the Output Control Register is specified as below:

Output Control Register A0 (*[T32AxOUTCRA0]*)
    Operation of T32AxOUTA <OCRA[1:0]>        No change
Output Control Register A1(*[T32AxOUTCRA1]*)
    Operation when the counter matches *[T32AREGA0]*.
    <OCRCMPA0[1:0]>                Set ("High")
    Operation when the counter matches *[T32AREGA1]*.
    <OCRCMPA1[1:0]>                Clear ("Low")

**Figure 5.4   Example of PPG Output (Low Active)**

While a duty ratio is changes at constant period, when T32AxOUTA is controlled, if a duty ratio unexpectedly exceeds the setting value or "0", the following operation will be led.

When a duty ratio is equal to a period (period A), the conditions of set and clear of the timer output are established at the same time. If multiple factors are established, T32AxOUTA is not changed.

When the "Low period" = 0 (period B), the clear and set conditions are established at the same time. But T32AxOUTA is set with priority since *[T32AxRGA0]* is the same value as the reload value.
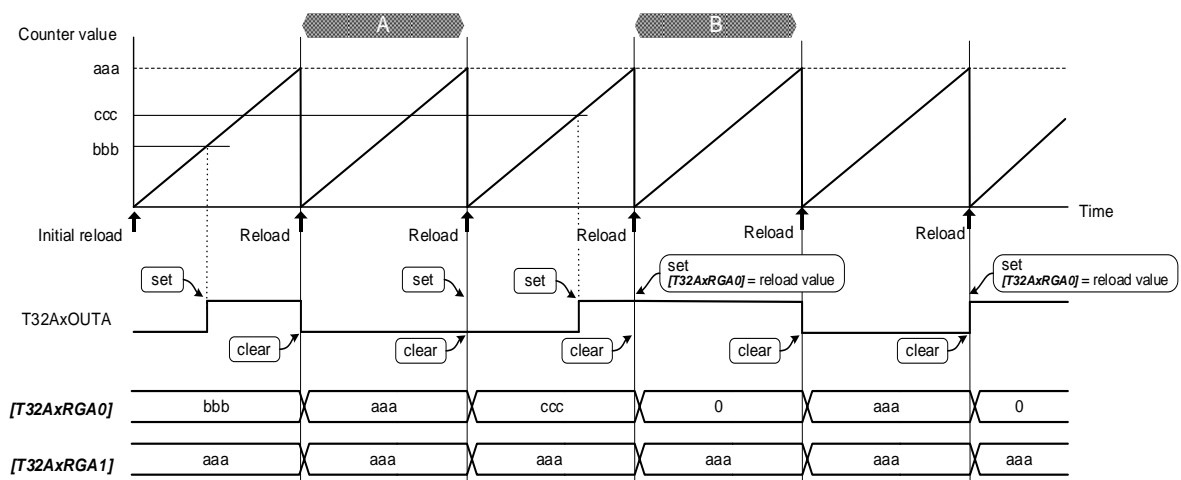
**Figure 5.5   PPG Output (Example of Period A, B)**

## 5.4. Programmable Rectangular Wave Output (PPG) by External Trigger

By starting a count with an external trigger, PPG with short delay time, which is too short to be handled by program can be output.

Counter Control Register A (*[T32AxCRA]*)
  Clock selection <CLKA[2:0]>      Any clock
  Count operation <UPDNA[1:0]>    Up counting
  Start operation <STARTA[2:0]>   On the rising edge of the external trigger
  Stop operation <STOPA[2:0]>    Any timing
  Reload Condition <RELDA[2:0]>   A match with the Timer Register A1(*[T32AxRGA1]*)
                  (Reloaded "0x0000")

The figure shows a waveform when the Output Control Register is specified as below:

Output Control Register A0 (*[T32AxOUTCRA0]*)
  Operation of T32AxOUTA <OCRA[1:0]>     No change
Output Control Register A0 (*[T32AxOUTCRA1]*)
  Operation when the counter matches *[T32AxRGA0]*.
   <OCRCMPA0[1:0]>        Set ("High")
  Operation when the counter matches *[T32AxRGA1]*.
   <OCRCMPA1[1:0]>        Clear ("Low")



**Figure 5.6  Example of PPG Output by External Trigger**

A delay time "aaa" from the external trigger is specified to *[T32AxRGA0]*. The added value "aaa" + "bbb" of a delay time "aaa" and a one-shot pulse width "bbb" is specified to *[T32AxRGA1]*.

At this time, if an external trigger pulse is input to T32AxINA0, the counter starts on the rising edge of the external trigger. When the counter value is incremented to "aaa", the counter value matches with *[T32AxRGA0]*. This allows T32AxOUTA to be "High" level. When the counter counts up until the value is "aaa" + "bbb", the counter value matches with *[T32AxRGA1]*. This clears T32AxOUTA and allows T32AxOUTA to be "Low" level.

## 5.5. PPG Outputs by Synchronous Operation

PPG can output programmable rectangular signals that are different duty ratio at the same frequency, by combining multiple timers. In this case, Timer A is used as the master, and Timer B is used as the slave.

The following is the definition of the period and duty ratio.

| | Period | Duty ratio (Timer A output) | Duty ratio (Timer B output) |
|---|---|---|---|
| Up counting | *[T32AxRGA1] – [T32AxRELDA]* | *[T32AxRGA0] – [T32AxRELDA]* | *[T32AxRGB0] – [T32AxRELDB] +[T32AxRGA1] – [T32AxRGB1]* |

Set Timer A to output a programmable rectangular wave. When the counter of timer A reaches "aaa", it matches with *[T32AxRGA0]* and then the level of T32AxOUTA are reversed. When the counter reaches "bbb", it matches with *[T32AxRGA1]* and then counter value is reloaded and counter is continue counting up.

Set the conditions of start, stop, and reload of Timer B to synchronous operation. Set the value to output a desired rectangular wave to the Timer Register B0/B1 *[T32AxRGB0/ T32AxRGB1]*. (set "ccc" to *[T32AxRGB0]*, and set "ddd" to *[T32AxRGB1]*) Set the reload value to the same value as the reload value of Timer A.

Counter Control Register A (*[T32AxCRA]*)
 Clock selection <CLKA[2:0]>     Any clock
 Count operation <UPDNA[1:0]>   Up counting
 Start operation <STARTA[2:0]>   Any timing
 Stop operation <STOPA[2:0]>   Any timing
 Reload condition <RELDA[2:0]>   A match with *[T32AxRGA1]* ("0x0000" is reloaded.)

Timer Register A0/A1
 Timer Register A0 (*[T32AxRGA0]*)   "aaa"
 Timer Register A1 (*[T32AxRGA1]*)   "bbb"

Output Control Register A1 (*[T32AxOUTCRA1]*)
 Operation when the counter matches *[T32AxRGA0]*.
  <OCRCMPA0[1:0]>         Set ("High")
 Operation when the counter matches *[T32AxRGA1]*.
  <OCRCMPA1[1:0]>         Clear ("Low")

Counter Control Register B (*[T32AxCRB]*)
 Clock selection <CLKB[2:0]>     Same as Timer A
 Count operation <UPDNB[1:0]>   Up counting
 Start operation <STARTB[2:0]>   Synchronous operation
 Stop operation <STOPB[2:0]>   Synchronous operation
 Reload Condition <RELDB[2:0]>   Synchronous operation

Timer Register B0/B1
 Timer Register B0 (*[T32AxRGB0]*)   "ccc"
 Timer Register B1 (*[T32AxRGB1]*)   "ddd"

Output Control Register B1 (*[T32AxOUTCRB1]*)
 Operation when the counter matches *[T32AxRGB0]*.
  <OCRCMPB0[1:0]>         Set ("High")
 Operation when the counter matches *[T32AxRGB1]*.
  <OCRCMPB1[1:0]>         Clear ("Low")
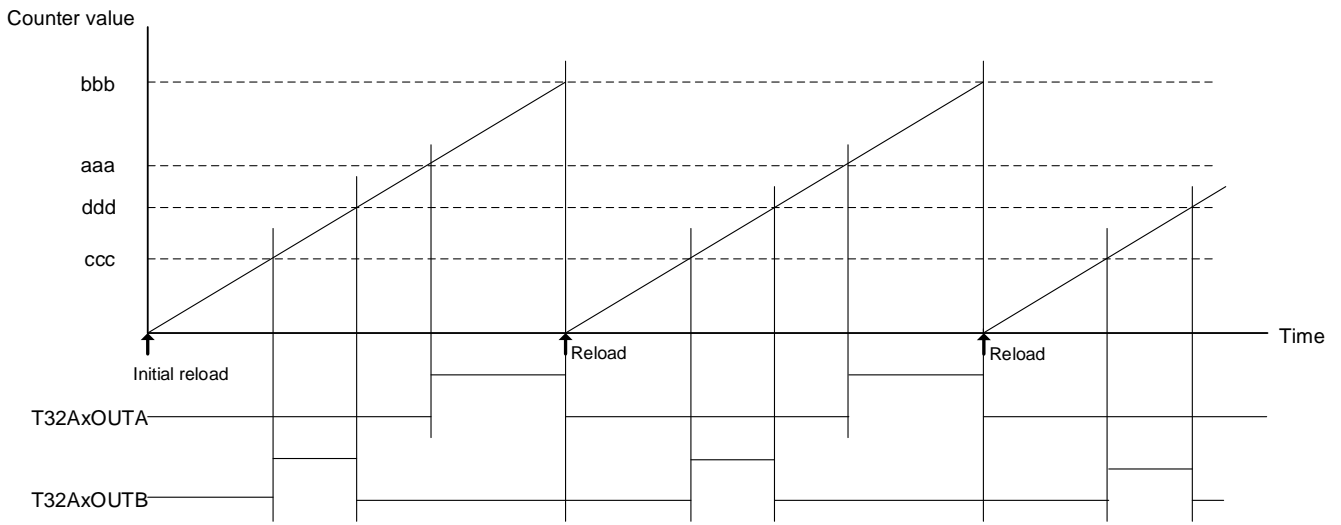
**Figure 5.7   PPG Outputs by Synchronous Operation (Low Active)**

## 5.6. Control Stepping Motor by PPG Outputs

PPG outputs control a stepping motor using the combination of two timers. In this example, PPG outputs are used in 32-bit timer. It uses internal trigger for timer stop condition. The internal trigger cannot be used depending on the product. Refer to "Product Information" of reference manual.

One timer (channel 1) is used to output a rectangular wave; the another timer (channel 0) is used to control the timer (channel 1). The counter value of channel 0 indicates the specific position information.

Channel 0 uses the rectangular output from channel 1 as the count clock. It counts on the falling edge of the signal from channel 1. After the reload value (bbb) and the compare match value (aaa) are set, perform the soft start.

Start the operation of channel 1 at the arbitrary conditions. Connect an internal trigger of channel 1 to the match detection circuit of channel 0 to stop the channel 1 at this condition.

When a rectangular wave of channel 1 is output, channel 0 starts down counting from the reloaded value until the counter is "aaa" to output an interrupt. Channel 1 detects a compare match as an internal trigger and stops operation. Since a rectangular wave is stopped outputting, channel 0 stops operation as well.

Change the setting of ch0 to count up and set the value (ccc) used for a compare match.

When channel 1 restarts to output a rectangular wave, channel 0 starts up counting until the counter is "ccc" to output an interrupt.

Counter Control Register C (*[T32AxCRC]*) (channel 0)
    Clock selection <CLKC[2:0]>               On the falling edge of an output from other timer
                                                 (T32AxTRGINCPHCK)
                                                    (rectangular wave output of channel 1)
    Count operation<UPDNC[1:0]>        Down counting
                                                (switching to up counting during the count operation)
    Start operation <STARTC[2:0]>       No trigger is used
    Stop operation <STOPC[2:0]>        Any timing
    Reload Condition <RELDC[2:0]>      None (free-running)

RUN Register C (*[T32AxRUNC]*) (channel 0)
    Start operation <SFTSTAC>          Start the counter operation by the program.

Counter Control Register C (*[T32AxCRC]*) (channel 1)
    Clock selection <CLKC[2:0]>               Any clock
    Count Operation <UPDNC[1:0]>       Any count operation
    Start operation <STARTC[2:0]>       Any timing
    Stop operation <STOPC[2:0]>        Internal trigger
                                                 (a match detection of channel 0)
    Reload Condition <RELDC[2:0]>      Any condition
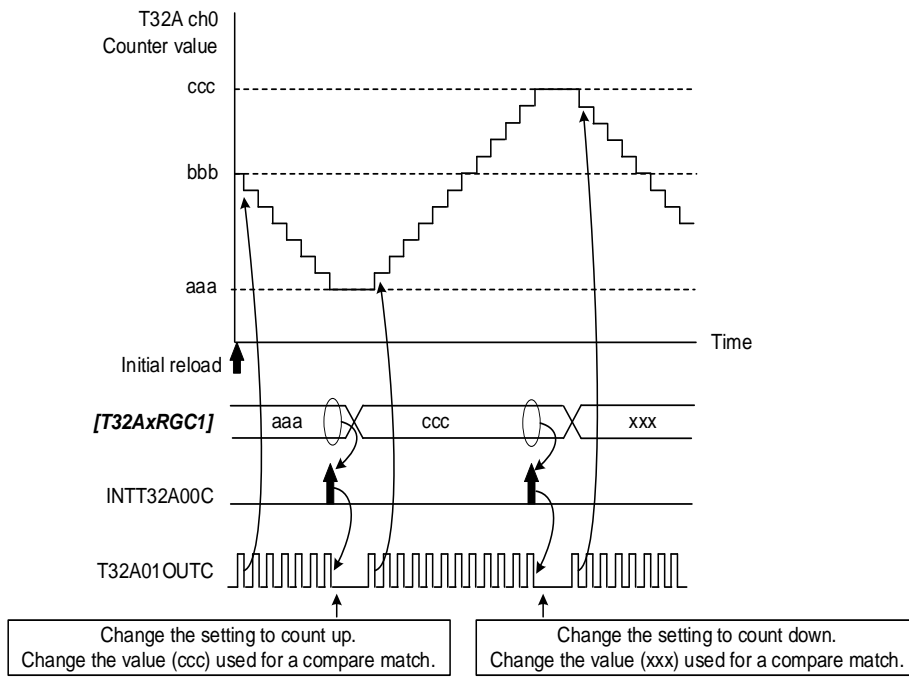
**Figure 5.8   PPG Outputs by Synchronous Operation (Example of Control of Stepping Motor)**

## 5.7. Frequency Measurement Using Capture Function

The following case is an example when a clock frequency input from an external source is measured.

In this case, Timer B is used as an interval timer; Timer A is used for outputting a programmable rectangular wave.

Use the external trigger (T32AxINB0) as the count clock for Timer B. Timer B counts up as a free-running counter.

Specify Timer A to generate the pulses (T32AxOUTA) for measurement time. Connect Timer A to an output from other timer at Timer B (T32AxTRGINBPHCK).

Specify the Capture Control Register *[T32AxCAPCRB]* of Timer B to capture the counter value to *[T32AxCAPB0]* on the rising edge of an output from other timer and to capture the counter value to *[T32AxCAPB1]* on the falling edge of an output from other timer.

When the counter of Timer A matches with *[T32AxRGA0]*, T32AxOUTA rises. The counter value of Timer B is captured to *[T32AxCAPB0]*. When the counter of Timer A matches with *[T32AxRGA1]*, T32AxOUTA falls. The counter value of Timer B is captured to *[T32AxCAPB1]*.

A frequency is determined by dividing (*[T32AxCAPB1]*-*[T32AxCAPB0]*) by the pulse width of T32AxOUTA.

 

Counter Control Register A (*[T32AxCRB]*)
    Clock selection <CLKB[2:0]>           External trigger
    Count operation <UPDNB[1:0]>      Up counting
    Start operation <STARTB[2:0]>      Any timing
    Stop operation <STOPB[2:0]>       Any timing
    Reload condition <RELDB[2:0]>      Any condition (free-running)

 

Capture Control Register B (*[T32AxCAPCRB]*)
    Capture timing of *[T32AxCAPB0]* <CAPMB0[2:0]>    On the rising edge of an output from other timer<T32AxTRGINBPHCK>
    Capture timing of *[T32AxCAPB1]* <CAPMB1[2:0]>    On the falling edge of an output from other timer<T32AxTRGINBPHCK>

 

Counter Control Register A (*[T32AxCRA]*)
    Clock selection <CLKA[2:0]>           Any clock
    Count operation <UPDNA[1:0]>      Up counting
    Start operation <STARTA[2:0]>      Any timing
    Stop operation <STOPA[2:0]>       Any timing
    Reload condition <RELDA[2:0]>      A match with Timer Register A1 (Reloaded "0x0000")

 

Timer Register A0/A1
    Timer Register A0 (*[T32AxRGA0]*)      Set the value to generate the pulse for the measurement time.
    Timer Register A1 (*[T32AxRGA1]*)

 

Output Control Register A1 (*[T32AxOUTCRA1]*)
    Operation of T32AxOUTA <OCRA[1:0]>          No change
    Operation when the counter matches with *[T32AxRGA0]*.
    <OCRCMPA0[1:0]>                   Set ("High")
    Operation when the counter matches with *[T32AxRGA1]*.
    <OCRCMPA1[1:0]>                   Clear ("Low")
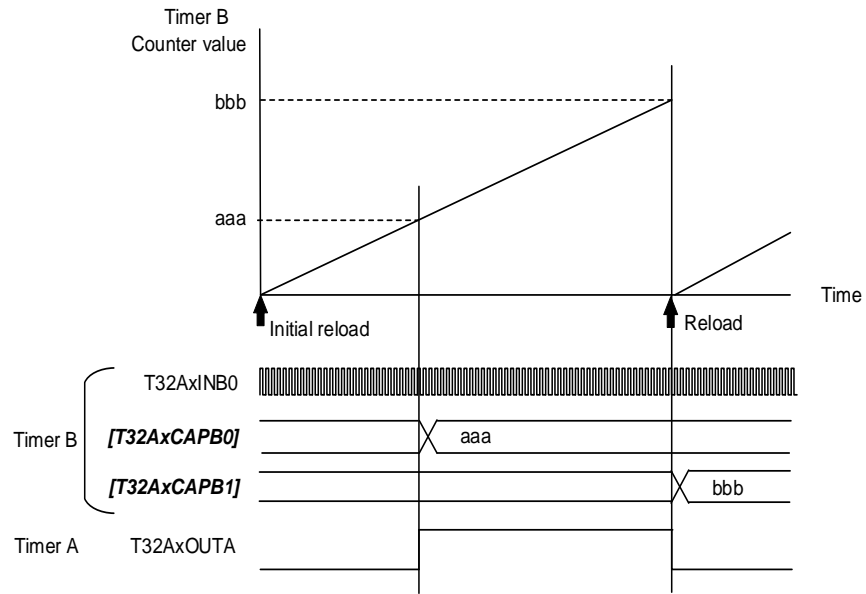
**Figure 5.9　Frequency Measurement Using Capture Function**

## 5.8. Pulse Width Measurement Using Capture Function

The following case is an example to measure the "High" level pulse width of input from an external source.

With *[T32AxCAPCRA]*, the counter value is captured to *[T32AxCAPA0]* on the rising edge of the T32AxINA0 pin and to *[T32AxCAPA1]* on the falling edge of the T32AxINA0.

When a rising signal of the external pulse is input to the T32AxINA0 pin, the counter value is captured to *[T32AxCAPA0]*. When a falling signal of external pulse is input to T32AxINA0, the counter value is captured to *[T32AxCAPA1]*.

The "High" level width of the external pulse can be determined the difference between *[T32AxCAPA1]* and *[T32AxCAPA0]*; then, multiplying the difference by the clock cycle of the prescaler output.

```
Counter Control Register A ([T32AxCRA])
    Clock selection <CLKA[2:0]>            Prescaler output
    Count Operation <UPDNA[1:0]>          Up counting
    Start operation <STARTA[2:0]>         Any timing
    Stop operation <STOPA[2:0]>           Any timing
    Reload condition <RELDA[2:0]>         Any condition


Capture Control Register A ([T32AxCAPCRA])
    Capture timing of [T32AxCAPA0].
    <CAPMA0[2:0]>                         On the rising edge of T32AxINA0
    Capture timing of [T32AxCAPA1].
    <CAPMA1[2:0]>                         On the falling edge of T32AxINA0
```

**Figure 5.10 Pulse Width Measurement Using Capture Function**

## 5.9. Time Difference Measurement Using Capture Function

The following case is an example to measure the time difference between the pulses input from an external source.

With *[T32AxCAPCRA]*, capture the counter value to *[T32AxCAPA0]* on the rising edge of the T32AxINA0 pin and to *[T32AxCAPA1]* on the falling edge of T32AxINA1 pin.

A time difference is determined by subtracting *[T32AxCAPA0]* from *[T32AxCAPA1]*; then multiplying the calculated value by the clock cycle of the prescaler output.

Counter Control Register A (*[T32AxCRA]*)
Clock selection <CLKA[2:0]>         Prescaler output
Count operation <UPDNA[1:0]>     Up counting
Start operation <STARTA[2:0]>    Any timing
Stop operation <STOPA[2:0]>     Any timing
Reload condition <RELDA[2:0]>    Any condition

Capture Control Register A (*[T32AxCAPCRA]*)
Capture timing of *[T32AxCAPA0]*.
<CAPMA0[2:0]>             On the rising edge of T32AxINA0
Capture timing of *[T32AxCAPA1]*.
<CAPMA1[2:0]>             On the rising edge of T32AxINA1
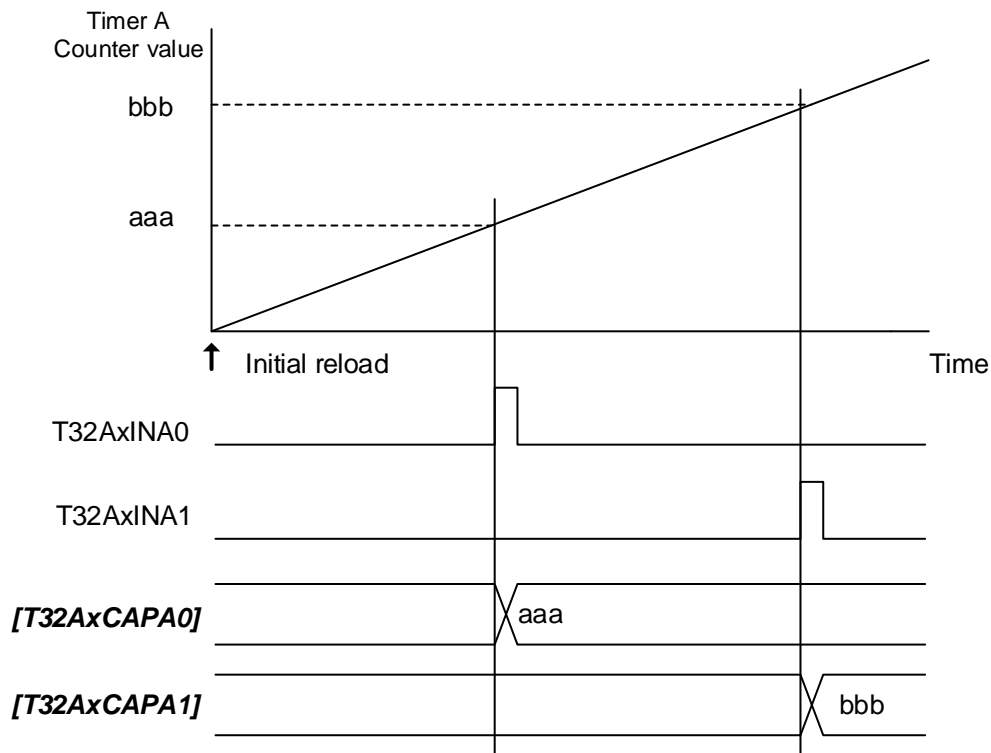


**Figure 5.11　Time Difference Measurement Using Capture Function**

# 6. Revision History

**Table 6.1   Revision History**

| Revision | Date | Description |
|---|---|---|
| 1.0 | 2020-12-24 | First release |
| 1.1 | 2022-06-30 | - 4.1. List of Registers<br> Added note to Timer A register.<br> Added note to Timer B register.<br> Changed note of Timer C register.<br>- 4.3.15. [T32AxCPA0] (Compare Register A0)<br> Added note.<br>- 4.3.16. [T32AxCPA1] (Compare Register A1)<br> Added note.<br>- 4.4.15. [T32AxCPB0] (Compare Register B0)<br> Added note.<br>- 4.4.16. [T32AxCPB1] (Compare Register B1)<br> Added note.<br>- 4.5.16. [T32AxCPC0] (Compare Register C0)<br> Added note.<br>- 4.5.17. [T32AxCPC1] (Compare Register C1)<br> Added note. |
| 1.2 | 2024-11-29 | - Appearance updated<br>- 2.1. 16-bit Timer<br> Changed table 2.1<br>- 2.2. 32-bit Timer<br> Changed table 2.2<br>- 3.8. Output Programmable Rectangular Wave (PPG)<br>      (T32AxOUTA/T32AxOUTB/T32AxOUTC)<br> Changed description<br>- 4.1. List of Registers<br> Changed note of (3) Timer A register<br> Changed table and note of (4) Timer B register<br> Changed note of (5) Timer C register<br>- 4.3.15. *[T32AxCPA0]* (Compare Register A0)<br> Changed note<br>- 4.3.16. *[T32AxCPA1]* (Compare Register A1)<br> Changed note<br>- 4.4.15. *[T32AxCPB0]* (Compare Register B0)<br> Changed note<br>- 4.4.16. *[T32AxCPB1]* (Compare Register B1)<br> Changed note<br>- 4.5.16. *[T32AxCPC0]* (Compare Register C0)<br> Changed note<br>- 4.5.17. *[T32AxCPC1]* (Compare Register C1)<br> Changed note<br>- 5.3. Programmable Rectangular Wave Output (PPG)<br> Changed description<br> Changed figure 5.5 |

## RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as "TOSHIBA".
Hardware, software and systems described in this document are collectively referred to as "Product".

## Toshiba Electronic Devices & Storage Corporation

https://toshiba.semicon-storage.com/